QL Today

Volume 7
Issue 4
Nov./December
2002

ISSN 1432-5454

The Magazine about QL, QDOS, Sinclair Computers, SMSQ...

www.QLToday.com

Contents

Advertisers

in alphabetical order

QL *Today*

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
Mobile +49 1701 222277
email: JMerz@j-m-s.com
email: QLToday@j-m-s.com

English office:

Q Branch
20 Locks Hill
Portslade
BN41 2LB
United Kingdom

Tel. +44 1273 386030
Mobile +44 7836 745501
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk
email: QLToday@j-m-s.com

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Tel. +44 1248 354023
Fax +44 1248 354023
email: dilwyn.jones@tesco.net
email: QLToday@J-M-S.com

Co-Editor:

Bruce Nicholls
38 Derham Gardens
Upminster
Essex RM14 3HA
United Kingdom

Tel +44 1708 510764
Fax +44 870 0568755
email: qltoday@q-v-d.demon.co.uk
email: QLToday@j-m-s.com

QL *Today* is published bi-monthly, our volume begins on beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL *Today*** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April	Issue 2: 30 June
Issue 3: 30 August	Issue 4: 30 October
Issue 5: 30 December	Issue 6: 28 February

QL *Today* reserves the right to publish or not publish any material submitted. Under no circumstances will **QL *Today*** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL *Today***. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

Editorial

Dilwyn Jones

First of all, please accept our apologies if this issue arrives late. We ran into a number of problems, for example we had planned a rather special cover disk (the first Cover CD by a dedicated QL magazine we think!) but we've had to give in and postpone it until the next issue.

At one stage we were short of articles, then following a special request we ended up with more than we have space for. Therefore, our apologies to those whose articles have not been used this time, they should be in the next issue instead.

The activity recently has focused mainly on the SMSQ/E sources project and the work done by Marcel Kilgus on updating the Window Manager (more details elsewhere in this issue). As the "colour drivers" have been around for a while (indeed there are a surprising number of applications which use GD2 to give the proper name, as I found out when I went to list those I knew of!) it seems appropriate that the Window Manager is now being brought up to date in line.

The Q60 has been making quiet progress too, with the latest release of the 68K Linux CD which has perhaps received less publicity than the above projects.

Rich Mellor is trying to assess how many people might be interested in purchasing replacement QL keyboard membranes if an attempt were to be made to produce another batch. Enquiries are being made in Europe and America to see how practical and economical it might be to produce more membranes. Please contact Rich Mellor at RWAP Software if you might be interested in buying a membrane. The more interest, the better the chances that a new batch might be manufactured.

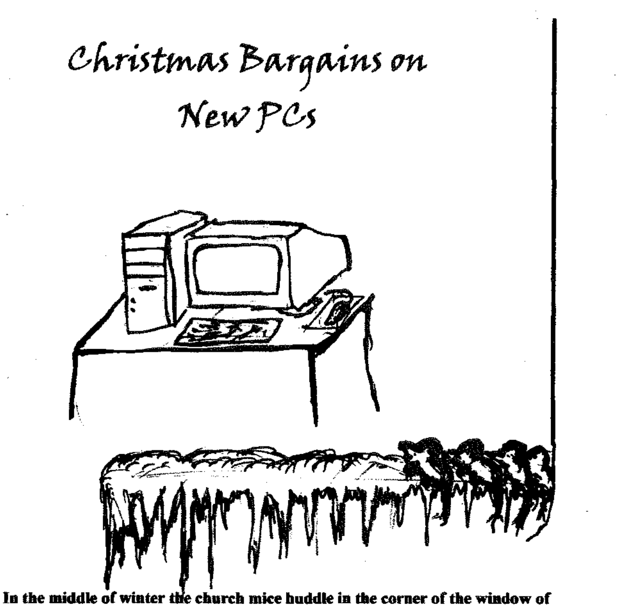
On the software front, Jim Hunkins is most certainly making progress with the QDT QL Desktop system, with occasional pictures and details escaping from him! Not that I like to mention other computers in these pages, but Jim Hunkins does work with well-known computers famed for their GUIs and graphical applications, so expect something special! Whether you like GUIs (Graphical User Interfaces) or not, most computers have them and it will become more and more important to at least have the choice of using one. There was a demonstration of an early version of QDT at the Byfleet Quanta workshop a few weeks ago.

You will note the distinct lack of QL shows in the near future! If you know of any entries for our show calendar please let us know. At the time of writing this, Quanta were inviting interested sub-groups to submit bids to host the Annual General Meeting, usually held on the third or fourth Sunday in April. If you'd like to see it in your area, contact your nearest Quanta sub-group and persuade them to apply to host the event!

It will not be long before we have to start thinking about what to do to celebrate 20 years of the QL, since it first appeared 1983/4. How do you think we should celebrate 20 years of QDOS? Write and let us know.

Finally, many thanks to our loyal subscribers and authors, who make QL Today possible. Our best wishes to you for Christmas QLing and a very happy new year to you all!

Cartoon



In the middle of winter the church mice huddle in the corner of the window of the local computer shop. They gaze in wonder and envy at the luxury of the computer mice in bright air conditioned splendour each with his own mat.....

NEWS

RWAP News, New Address

Please note that I have moved. My new address details are set out at the end of this news item.

I may be out of touch for a little while whilst I get settled in.

My website is always worth a visit - we have now added our own guestbook and visitor counters for you to try.

Keep up the good work everyone...

Rich Mellor
RWAP Software
35 Chantry Croft,
Kinsley,
Pontefract,
West Yorkshire,
WF9 5JH
Tel: 01977 6140509
http://hometown.aol.co.uk/rwapsoftware

QL PASCAL Manual

Jean-Yves Rouffiac writes:

I have begun putting scanned images of the Computer One Pascal user guide/manual on my web site. At present I have scanned enough pages for someone who already knows Pascal to get going as I have included the introduction, installation procedures and QL-specific functions (eg memory, graphics/windowing and file handling). The quality of the scans is readable but not great, based on a compromise between quality and file size. I'll have a go at an OCR solution too to see if that's feasible.

www.westhaven.uklinux.net/qwertyb/

Also, the Dreamlands text adventure is updated to cater for the WIN1_ device (at long last!). Unfortunately, as I now use Qemulator rather than QLAY, there is a slight problem with QLiberator which I am looking into, and at present v10.23 only runs in SuperBASIC. Not too much of a problem on today's PC's I should imagine.

GWASL V1.8

The Gwasl assembler (Gwass Assembler Lite) from ***George Gwilt***, used by Norman Dunbar for his QL Today assembler articles, has been updated to v1.8 to fix a minor problem with the MOVEP instruction. The new version is available for download from Dilwyns website on

www.soft.net.uk/dj/software/other/other.html

Dilwyn Jones's Website

As my ISP is shortly pulling the plug on the Softnet Gold service on December 18th 2002, my email address and website address will have to change.

At the time of writing, I do not yet know the new address but will announce it as soon as possible and place the website on a new web address as soon as possible. Those of you who have links to my website on your sites, please take note.

New SMSQ/E Reseller

Welcome to Phoebus Dokos as a new reseller of SMSQ/E in north America. For further details, contact him direct via the email address

phoebus@dokos-gr.net
Phoebus R. Dokos,
941 Lilac Street Apt #1,
Indiana,
PA 15701-3340,
USA.
Tel +1 (724) 464 0199

QEYMAIL

At the time of writing, Dave Park is inviting suggestions for facilities to add to his Qeymail email package for the QL. It will be soql based but anything else is open to ideas!

dexter@spodmail.com

QL Service Manual

We have added the 'Sinclair QL Service Manual' ZIP and PDF file formats to our website at the url:

http://badared.com/QL/mans/qlsm/qlsm_en.html

Javier Guerra Sinclair QL Spanish Resources

http://badared.com/QL

QL Toolkits Keywords List

François Van Emelen's toolkit keywords list is now available from the QL Documentation page on my website, as a plain text file. There is an email link on the QL Documentation page if you wish to email any extra details or corrections to him for inclusion.

http://www.soft.net.uk/dj/qldocs/qldocs.html

(scroll down to the bottom of the page)

The plain text file itself is at

http://www.soft.net.uk/dj/qldocs/toolkit.txt

(warning: about 102KB to download!)

The QL Net links page now includes a pointer to the World Of Spectrum pages, from where you can find copies of the QL Manual and QL Service Manual in Adobe PDF format in the Technical Docs section.

By following the link from my QL Net page to Javier Guerra's Sinclair QL Spanish Resources page, you can also find a copy of the QL Service Manual in HTML format.
www.soft.net.uk/dj/index.html

PIC to BMP File Conversion

Jerome Grimbert writes:
I have a pic2bmp program which converts QL PIC files to Windows BMP files (including GD2 ones) that works fine on my Q40 and QPC (as well as with native mode 4 and mode 8!)
http://jgrimbert.free.fr/ql/download.html

MODE 32 / MODE 33 File Conversion

Mode 33 and Mode 32 graphics file conversion can be achieved with a useful little program from George Gwilt.

This program will translate mode 33 or 32 partial save areas to the other mode (and back againif required!). It is command line driven via simple command line switches.

Because the programming of the conversion is so extremely easy in assembler language with 68020+ instructions I used them. Thus TR_PSA_BIN needs a Q40, Q60, super gold card, QXL or eqivalent hardware with 68020 or later processor. This may rule out QPC for example as it emulates a 68000/68010 processor, and possibly other emulators too.

This program can be found on Dilwyn's website Other Software Page.

PICVIEW

This is a new pointer driven program able to view both QL screens and PIC files, in both QL colour modes and GD2 colour modes. Designed to be quick and simple to use, this viewer is freeware and available for download from the My Freeware page on my website.
www.soft.net.uk/dj/freeware/freeware.html

PQIV with Postscript

Claus Graf's Q40 website includes a new version of PQIV. PQIV is an Image Viewer for Pointer Environment (demands SMSQ/E 2.95 or later), PCX, PNG, PIC, TIFF, GIF and JPEG support, saves as PIC or Background Image), prints to PostScript files, a Printer Dialog window lets you adjust the size of the image and place the image on the paper. You then need GhostScript to print to a huge variety of printers, which is freely available for SMS. A Rotate function lets you change the orientation of images.

The site also includes links to details of the Linux 2 Alpha (including Java!) for Q40/Q60 systems, for those who use Linux on these computers.

The site also includes software for Fast BASIC Floating Point operations with FPUFNs, by Simon Goodwin.

Claus Graf's website is on
www.q40.de

New Version of Turbo

A new version of Turbo is now available which allows programs to be compiled with machine code extensions attached to the body of the program, in much the same way as the same can be achieved with QLiberator. V4.16 of Turbo is from George Gwilt and may be downloaded from John Sadler's Scottish QL Users Group (SQLUG) website on
www.jms1.supanet.com
and from the Other Software Page on
www.soft.net.uk/dj/software/other/other.html
It will also be available from Dilwyn's PD library service. To go with this version of Turbo, version 3.33 of Turbo Toolkit is available from the same websites, along with the latest updates of the Turbo manuals.

Small Ads

FOR SALE: QL with green screen Ferguson monitor with software such as Scrabble, Abacus, Quill and a few games. Offers invited by **Jeremy Saunders, 4 Mearns Walk, Stonehaven, Kincardineshire, Scotland AB39 2DG. Telephone 01569-763752**, or email me on **jcl.saunders@virgin.net**

FOR SALE: Epson 850 Colour Inkjet Printer (fully QL compatible) £70 plus post and packing.
Rich Mellor RWAP Software, new address details on opposite page!

FOR SALE: Q40i System (built June 2002) 68040, 40Gb Hard Disk, 16Mb RAM, Keyboard, 15" Monitor, 1 x 1.44Mb FDD, Software Bundle (inc SMSQ/E), Keyboard and Mouse £499
AdventAthlon 1700XP, 40Gb Hard Disk, 256Mb RAM, Graphics Card, 56KModem, CD-ROM, 15", 1 x 1.44Mb FDD, LexmarkColour Printer, Keyboard, Mouse, QPC 2 v3, Windows 98 (inc CD),Office 2000 (inc CD), Speakers £499
Paul Merdinian: 01303 220924 (Folkestone) Flat 6, 48 Cheriton Road, Folkestone, Kent, CT20 1DD

New Window Manager?

Marcel Kilgus, not being content with the brilliant work he has been doing with QPC2, has also been hard at work implementing a new version of good old WMAN, the pointer environment Window Manager which gives that standard QL PE look and feel to the menus of many QL programs.

In mid-November, Marcel sent an email to the QL-users mailing list which described some of the work he had done at that time, and explaining that due to his college commitments there would be times when he'd be unable to work on it. He also gave a list of some system mnemonics and other information of interest to programmers.

Marcel wrote:
All things I wanted to integrate into WMAN are now there, though some still need to be tested. Unfortunately I'm running out of time and will probably have to (almost) stop SMSQ related work for a few weeks. For testing purposes I have also created a system-palette QPAC2. This at first looks exactly like the ordinary white/green windows. But when the palette gets changed to some other scheme it can for example look like this:
http://www.kilgus.net/qpc/qpac2.png

(We did not try to reproduce this in QL Today as black and white would not do it justice).
Still at an early stage, but already nice, I think. The mouse pointer there is my mode64 test pointer (clone of my windows pointer as I'm not good at that stuff).
If some programmers want to play with the new features I can probably build a test version, but QPC only at the moment. The development state is too early to be integrated into the other versions. That will probably follow in December.

```
Vector $7C                                         WM.SETSP

       Set system palette entries

Call parameters                                    Return parameters

D1.w start index                                   D1   preserved
D2.w number of elements                            D2   preserved
                                                   D3+  all preserved

A0                                                 A0   preserved
A1     pointer to palette entries / 0              A1   preserved
A2                                                 A2   preserved
A3                                                 A3   preserved
A4                                                 A4   preserved
A5     not used by any routine
A6     not used by any routine

Error returns:
       IPAR    Illegal index number / invalid number of elements
```

Set the entries of the system palette to the values in the buffer, beginning with the index in D1 (counting from 0) and ending with the index D1 + D2 - 1.

If A1 = 0 then the entries are taken out of the default table. Otherwise the buffer must hold an array of words with the colour values of the different items. The colour format is the standard WMAN colour format as described elsewhere.

```
Vector $80                                         WM.GETSP

       Read system palette entries

Call parameters                                    Return parameters

D1.w start index / -1                              D1.w preserved / item count
D2.w number of elements                            D2   preserved
                                                   D3+  all preserved

A0                                                 A0   preserved
A1     pointer to entry buffer                     A1   preserved
A2                                                 A2   preserved
A3                                                 A3   preserved
A4                                                 A4   preserved
A5     not used by any routine
A6     not used by any routine

Error returns:
       IPAR    Illegal index number / invalid number of elements
```

Copies entries of the system palette into the given buffer, beginning with the index in D1 (counting from 0) and ending with the index D1 + D2 - 1. The buffer must be big enough to hold all requested entries.

If D1 is given as -1 the function just returns the number of items held in the system palette. This can increase when more items get defined in new WMAN version. This is guaranteed to be below 256.

Since yesterday all routines accept the new colour word definition, only a few more lines for the system palette are needed. And some sensible defaults for that palette (which'll probably be the hardest task).

I'll attach the new WMAN specification the way I did implement it.

Any suggestions for changes should be done exactly NOW before it gets official!

New colour format:

```
%00000000cccccccc       exactly as before
%00000001pppppppp       palette
%00000010pppppppp       system palette
%00000011gggggggg       gray scale
%01ssxxxxxxyyyyyy       palette stipple. see below
%1rrrrrgggggbbbbb       15 bit RGB
```

Stipple format (as proposed by George Gwilt):
s = stipple code (0 = dot, 1 = horizontal, 2 = vertical, 3 = checkers)
x = stipple colour
y = main colour

As x and y can only hold 6 bit only the first 64 entries of the palette can be used for stippling. Due to the design of the palette those entries alone still cover the whole colour range quite well.

Finally, for anybody interested I have attached the current system palette. That should be almost the final version but I am of course still open for suggestions.

```
sp.winbd        equ     $0200   ; Window border
sp.winbg        equ     $0201   ; Window background
sp.winfg        equ     $0202   ; Window foreground
sp.titlebg      equ     $0203   ; Title background
sp.titletextbg  equ     $0204   ; Title text background
sp.titlefg      equ     $0205   ; Title foreground
sp.litemhigh    equ     $0206   ; Loose item highlight
sp.litemavabg   equ     $0207   ; Loose item available background
sp.litemavafg   equ     $0208   ; Loose item available foreground
sp.litemselbg   equ     $0209   ; Loose item selected background
sp.litemselfg   equ     $020a   ; Loose item selected foreground
sp.litemunabg   equ     $020b   ; Loose item unavailable background
sp.litemunafg   equ     $020c   ; Loose item unavailable foreground
sp.infwinbd     equ     $020d   ; Information window border
sp.infwinbg     equ     $020e   ; Information window background
sp.infwinfg     equ     $020f   ; Information window foreground
sp.infwinmg     equ     $0210   ; Information window middleground
sp.subinfbd     equ     $0211   ; Subsidiary information window border
sp.subinfbg     equ     $0212   ; Subsidiary information window background
sp.subinffg     equ     $0213   ; Subsidiary information window foreground
sp.subinfmg     equ     $0214   ; Subsidiary information window middleground
sp.appbd        equ     $0215   ; Application window border
sp.appbg        equ     $0216   ; Application window background
sp.appfg        equ     $0217   ; Application window foreground
sp.apphigh      equ     $0218   ; Application window item highlight
sp.appiavabg    equ     $0219   ; Application window item available background
sp.appiavafg    equ     $021a   ; Application window item available foreground
sp.appiselbg    equ     $021b   ; Application window item selected background
sp.appiselfg    equ     $021c   ; Application window item selected foreground
sp.appiunabg    equ     $021d   ; Application window item unavailable background
sp.appiunafg    equ     $021e   ; Application window item unavailable foreground
sp.scrbar       equ     $021f   ; Pan/scroll bar
sp.scrbarsec    equ     $0220   ; Pan/scroll bar section
sp.scrbararr    equ     $0221   ; Pan/scroll bar arrow
sp.buthigh      equ     $0222   ; Button highlight
sp.butbd        equ     $0223   ; Button border
sp.butbg        equ     $0224   ; Button background
sp.butfg        equ     $0225   ; Button foreground
sp.hintbd       equ     $0226   ; Hint border
sp.hintbg       equ     $0227   ; Hint background
sp.hintfg       equ     $0228   ; Hint foreground
sp.errbg        equ     $0229   ; Error background
sp.errfg        equ     $022a   ; Error foreground
sp.shaded       equ     $022b   ; Shaded area
sp.3dobjbg      equ     $022c   ; 3D object background
sp.3dobjfg      equ     $022d   ; 3D object foreground
```

Gee Graphics! (on the QL?) - Part 31

H. L. Schaaf

"Crowding in the Neighborhoods."

This time we add to the menu 2 neighborhoods, more subsets of the Delaunay triangulation. One set is known as the Nearest neighborhood and the other as the Relative neigborhood.

The nearest neighbor graph only connects each point to its closest neighbor. The edges of the nearest neighbor graph are a subset of a Minimum Spanning tree, so we may have dis-connected graphs. But all the "dots" are connected to at least one other (nearest) dot.

The relative neighbor concept connects points that mutually do not have any other points closer. The relative neighbor graph is a subset of the Gabriel graph, and a Minimum Spanning tree is a subset of the relative neighbor graph.

It is probably easier to understand all of this by seeing some examples for yourself. Two more PROCedures "NearNeig" and "RelNeig" will do the job. We're running out of room in the prompt window, so I'll just squeeze in a few letters this time, using 'N' for the Nearest neighbor graph, and 'L' for the reLative neighbor graph.

Start with the collection we had in GG#30 and then merge in the listing "Neighbors_PROCs".

```
Listing Neighbors_PROCs
------------------------------------------

5835 PRINT #0;' [N]eighbor  (re[L]), [M]ST, ';cst$;
5976  IF ans$ == 'n' THEN INK 2 :NearNeig : choose_options
5977  IF ans$ == 'l' THEN INK 2 :RelNeig : choose_options
12190 :
12200 REMark Neighbors_PROCs, for GG#31, October 28, 2002
12210 :
12220 DEFine PROCedure RelNeig
12230  LOCal i,j
12240  FOR i = 1 TO DIMN(Dedg)
12250   test_val = 0
12260   p_1_x = P(Dedg(i,1),1)
12270   p_1_y = P(Dedg(i,1),2)
12280   p_2_x = P(Dedg(i,2),1)
12290   p_2_y = P(Dedg(i,2),2)
12300   test_rad = dist_btwn (p_1_x ,p_1_y, p_2_x, p_2_y)
12310   FOR j = 1 TO DIMN(P)
12320    IF ((j <> Dedg(i,1)) AND (j <> Dedg(i,2)))  THEN
12330     test_dis_1 = dist_btwn(p_1_x,p_1_y,P(j,1),P(j,2))
12340     test_dis_2 = dist_btwn(p_2_x,p_2_y,P(j,1),P(j,2))
12350    END IF
12360     IF (test_dis_1 < test_rad) AND (test_dis_2 < test_rad) THEN
12370      test_val = 0
12380      EXIT j
12390     ELSE
12400      test_val = 1
12410     END IF
12420   END FOR j
12430   IF (test_val) : show_edge(i)
12440  END FOR i
12450 END DEFine RelNeig
12460 :
12470 DEFine PROCedure NearNeig
12480 LOCal i,j
12490 FOR i = 1 TO DIMN(P)
12500  edge_count = 0
12510  FOR j = 1 TO DIMN(Dedg)
12520   IF Dedg(j,1) = i OR Dedg(j,2)= i :edge_count = edge_count+1
12530  END FOR j
12540  DIM edges_of_v(edge_count,2)
12550  edge_count = 0
12560  FOR j = 1 TO DIMN(Dedg)
12570   IF Dedg(j,1) = i OR Dedg(j,2)= i THEN
12580    p_1_x = P(Dedg(j,1),1)
12590    p_1_y = P(Dedg(j,1),2)
12600    p_2_x = P(Dedg(j,2),1)
```

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets. Can be used with lights/hifi/monitors—ie a QL monitor can be used as a switch control.

Cost £24

superHermes

A major hardware upgrade for the QL

All Hermes features (working serl/2 at 19200, independent baud rates/de-bounced keyboard/ keyclick) IBM AT kbd I/F // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software)	**£90** (£92/£93)
IBM AT UK layout Keyboard	**£11 (£**13/£15)
Serial mouse	**£8** (£8.50/£9)
Capslock/scrollock LED	**£1** (£1.50/£1.50)
Keyboard or mouse lead	**£3** (£3.50/£3.50)
High speed serial (ser3) lead	**£4** (£4.50/£4.50)

Hermes available for £25 (£26/£27) **Working serl/2 and independent input, debounced keyboard.**

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.
Cost (incl keyboard lead) ....**£53** (£54/£55)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if requd).
Send disk plus SAE or two IRCs

MKI..£40 (£41/£43) **MKII..£65** (£66/£67)

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I^{2}C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off **RomDisq** at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq	**£39** (£40/£41)
4mbytes RomDisq	**£65**(£66/£67)
8 mbytes RomDisq	**£98** (£99/£100)
Aurora adaptor	**£3** (£3.50/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost..**£34** (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I^{2}C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)
2 amp (for 8 relays, small motors)......................**£40** (£43/£44)
4 amp total (for motors etc)........................**£45** (£48/£50)
Relays (8 3a 12v 2-way mains relays (needs 2a power driver) ....**£25** (£28/£29)
Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required ....**£25** (£27/£28)
Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting..**£30** (£31/£32)
Temp probe (-40°C to +125°C) ....**£10** (£10.50/£11)
Connector for four temp probes ....**£10** (£10.50/£11)
Data sheets..**£2** (£2.50/£3)
Control software & manual (for all I/F).........**£2** (£2.50/£3)

QL SPARES

Keyboard membrane	**no longer on sale**
1377 PAL	**£3** (£3.50/£4)
Circuit diagrams	**£3** (£3.50/£4)
68008 cpu or 8049 IPC	**£8** (£8.50/£9)
8301/8302 or JM ROM or serial lead	**£10** (£10.50/£11)
Power supply (sea mail overseas)	**£12** (£17/£21)

Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable) Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address,/postal order or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details **31 Oct 02**

29 Longfield Road, TRING, Herts, HP23 4DG
Tel: 01442-828254 Fax/BBS: 01442-828255
tony@firshman.co.uk http://www.firshman.co.uk

```
12610    p_2_y = P(Dedg(j,2),2)
12620    edge_count = edge_count+1
12630    edges_of_v(edge_count,1) = j
12640    edges_of_v(edge_count,2) = dist_btwn( p_1_x, p_1_y, p_2_x, p_2_y)
12650   END IF
12660  END FOR j
12670  edges_of_v(0,2) = edges_of_v(1,2)
12680  edges_of_v(0,0) = edges_of_v(1,1)
12690  FOR j = 1 TO edge_count
12700  IF edges_of_v(j,2) < edges_of_v(0,2) THEN
12710   edges_of_v(0,2) = edges_of_v(j,2)
12720   edges_of_v(0,0) = edges_of_v(j,1)
12730  END IF
12740  END FOR j
12750  show_edge edges_of_v(0,0)
12760 END FOR i
12770 END DEFine NearNeig
12780 :
12790 REMark end of Neighbors_PROCs
```

Next time? I really don't know yet.

Digital Photography on your QL

Phil Stokes

I make no apologies about the lack of pictures in this article as my computer is thousands of miles away as I write this as I'm currently working in Afghanistan until December, even though it's all about taking and processing, and then printing Photos on your QL (QXL, QPC, Q40/60, Qlay, Aurora etc - I'll stick with plain QL from now on!) There are basically three stages from actually taking the picture, to having a hard copy (assuming that's what you want)

1. Taking the picture is easy, press the "shutter" button on the camera, getting it into your QL is the complicated part, there are several options here:

a. Cheat and use a PC/Mac for which any camera usually comes with leads and software, save the file to Floppy and then place the floppy into your "QL" and hey presto! - (Downside needing a PC!) in which case you may well use the thing to do the whole job, and you're by now reading the next article!

b. You need a QL compatible camera - Basically a Kodak camera with a serial port e.g. DC215 like mine or DC200 see Simon N Goodwins excellent article on Digital Cameras for a more accurate description, I would go into greater detail but he has already done this and he knows far more about them than I, and the article is back home (see intro!). Then use his excellent DIGICAM BAS software to download the pictures into your QL and save to MDV, FLP, WIN, ROM etc go to stage 2.

c. You need a "QL" with a compact flash drive e.g. Q60 or QPC with relevant hardware i.e. compact flash reader! and either drivers to read the card or a utility programme, in which case you probably know more than me! Anyway you still need the photos on file in an easily accessable way

2. You should by now have in your QL the Photos you want, probably in JPEG format, the problem being that you cannot view them directly! You need either photon or pqiv (or any other utility that works! that I've not heard of. If so then let the wider QL community know!). You then need to save a copy of the photo into a QL native format ie SCR or PIC. I believe PQIV can do this already but needs a late version of SMSQ/E, while this is planned for Photon (please hurry up) which will work with any QDOS configuration. The problem here being that the photo may well be bigger than your screen to just

```
SBYTES FLP1_Apic_PIC,131072,32768
```

naughty but works on some machines, apart from the fact that you only get part of the picture! I have written a routine to "sew" four screenshots back together but it needs changing as it currently only reassembles a 640x480 picture in mode 8 due to the maths and pixel shape changing between modes, but if your screen is big enough (resolution wise) then you can just save the relevant parts as a pic image, routine written for ALL modes but only tried on QL modes 4&8 until my Q60 appears (sometime after I order it!). Volunteers please wait until after I return home! But a way to save from Photon would be most appreciated!

Next all we need to do is load the picture into a bitmap editer programme, unfortunatly I know of no such programme currently availiable capable of editing pictures in modes except 4&8 and also of any size - memory permitting! So I've decided to write one, currently it is unfinished but working and capable of editing mode 4&8 up to a theoretical resolution of 32768 x 32768 which should keep us going for a while yet. Mode support is already built in for modes 33&32 in a lot of operations by simply using a variable for the number of bytes needed to store a pixel and limiting moves to factors of the groups of bytes in a group, needed in mode 4&8 to speed things up as two adjacent bytes hold the patterns for 8/4 pixels interleaved (nice Eh!) but all other mode don't (Phew!) according the the SuperBASIC reference manual (which Rich Mellor keeps splendidly up to date! I apologise if mine is slightly behind!). It is in theory possible to edit a picture in say mode 33 on an original QL memory permitting and I'll try to implement this! And it is certainly possible to convert between modes. This WILL be implemented in zoomed in Pixel editing, copying, moving pieces whilst filtering out certain colour ie chromacy (probably spelt wrong - used to create special effects on TV). Then the picture can be either saved or printed. Idealy it needs to be saved as a PIC image easier use!

3. Printing - another difficult subject! Either:

a. If you have a printer which is SDUMP compatible then the following sequence will do

```
pic=ALCHP(picture size)
LBYTES FLP1_PIC_PIC,pic
SDUMP pic
RECHP pic
```

assuming the you have aready set the BAUD rate and SDUMP parameters, see the manual

b. use the Prowess drivers. I don't know how to access them direct yet but Linedesign will do it for you! Simply use load bitmap, then in EDIT mode scale the picture as required, and Print page!

c. use Ghostscript ported/written(? certainly a lot of hard work!) by Johnathon Hudson which I am led to believe by the Q60 website supports a load of current printers and probably therefore needs a lot of memory but then so do graphical applications by there very nature the pictures are certainly bigger than my programme!

Clearly work still needs to be done but I've already got pictures with pints of beer all over them, who said the Camera never Lies! Certainly your QL & Digital camera can lie through their teeth as it were! My apologies for any inaccuracies in my article and any offences caused but due to my situation I hope you can understand why they arose. I have no ties to any commercial programmes mentioned which I use except as a happy user, and only wish to show you that your QL still has a future inside whatever box it resides!

PS. A very select few may have seen an earlier version of my programme at the last QUANTA show in Manchester to which I paid an all to brief visit on the Saturday! The show was otherwise very much worth a visit!

QDT Progress Report

Jim Hunkins

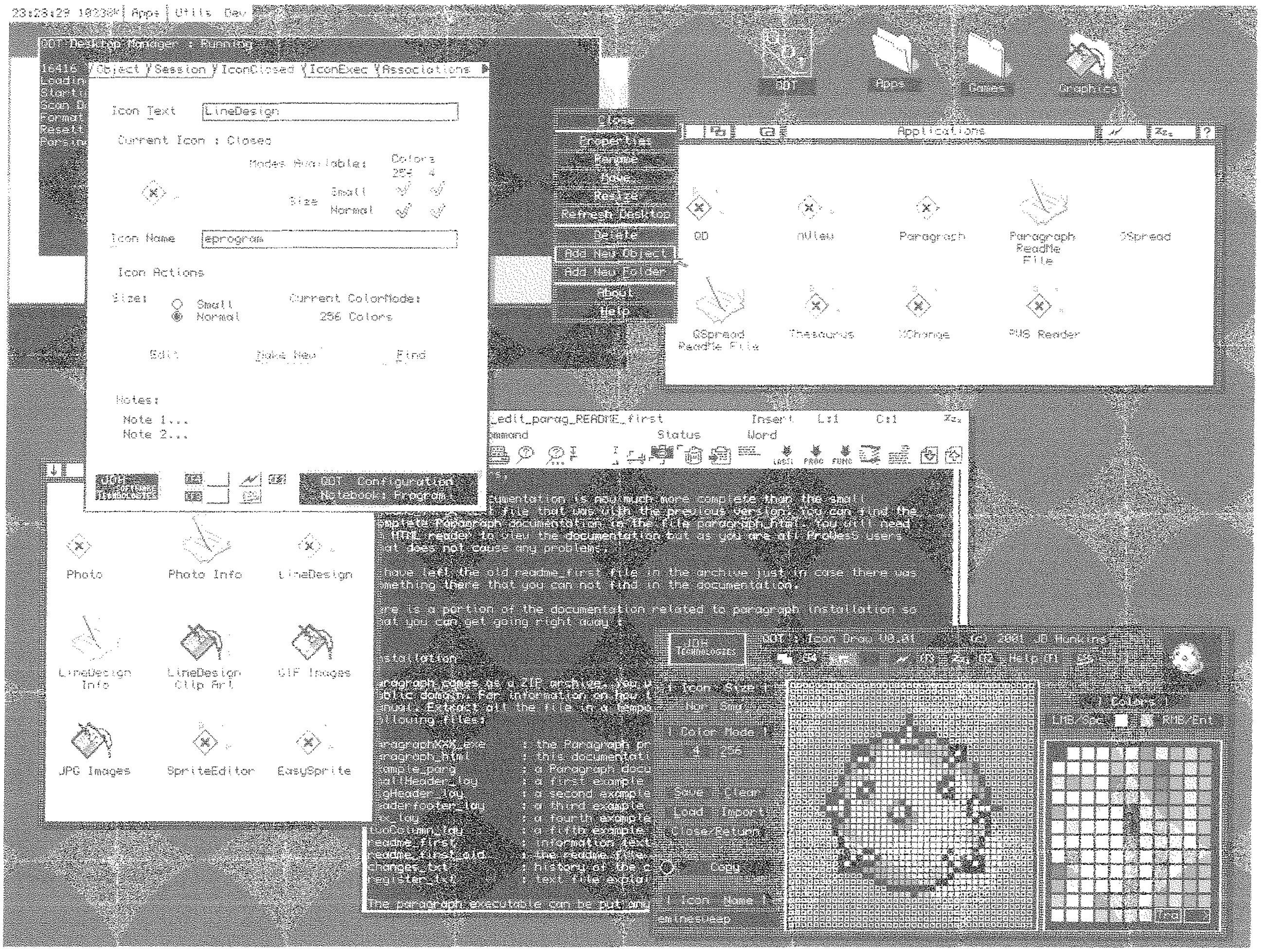
When working on something as complex as a modern desktop (IE: QDT), things just seem to trudge along. Working on so many individual pieces, you often feel like you aren't making much progress. And then, one day, you actually run the code from the very top and - WOW!

Progress QDT Desktop Snapshot

I have included a screen shot (see next page) showing QDT in action. What you see in the desktop screen capture is all real. The upper right hand shows four different desktop objects (icons); the QDT desktop itself and three separate folders. A subtle piece here is that the Apps folder is open (the actual open folder is seen directly under the objects) and the object shows an open folder. The Games folder on the other hand is closed and the object appropriately shows a closed folder. The Graphics folder object shows a case of incompleteness - the fact that I have only designed one icon for that folder so far which is used for both the open and closed view.

Under the traditional buttons (yes, QDT is designed to mix with all the other options out there for the SMSQ/E environment) is a text window that is showing the QDT activity - very useful during development and for debugging problems. This can also be turned off.

Over the top of that window is a page from the tabbed configuration notebook. This notebook is for the LineDesign program object in the open folder



```
_edit_parag_README_first              Insert    L:1     C:1
ommand                                Status    Word

cumentation is now much more complete than the small
t file that was with the previous version. You can find the
mplete Paragraph documentation in the file paragraph_html. You will need
HTML reader to view the documentation but as you are all ProWesS users
at does not cause any problems.

have left the old readme_first file in the archive just in case there was
mething there that you can not find in the documentation.

re is a portion of the documentation related to paragraph installation so
at you can get going right away :

stallation

ragraph comes as a ZIP archive. You w
blic domain. For information on how t
nual. Extract all the file in a tempo
llowing files:

ragraphXXX_exe         : the Paragraph pr
ragraph_html           : this documentati
ample_parg             : a Paragraph docu
allHeader_lay          : a first example
gHeader_lay            : a second example
aderfooter_lay         : a third example
x_lay                  : a fourth example
twoColumn_lay          : a fifth example
readme_first           : information text
readme_first_old       : the readme file
changes_txt            : history of the c
register_txt           : text file expla

The paragraph executable can be put any
```

directly under the notebook. The notebook shows the full use of the hi color mode in the actual PE window section (complements of Marcel's work on the PE and WMAN). I probably won't get around to actually tuning in the hi color for all the components until towards the end, even thought the objects all are fully hi-color by default now. Of course, as a user of QDT, you will have free reign to tune and tweak the colors to your heart's delight. And sometime late next year, you will even be able to share your desktop look and feel through the planned Theme Manager.

From this notebook, I opened the Icon Draw program to see if I could find and/or design a better icon for this object. You can see that program in the lower right hand side of the screen.

The text file directly behind the Icon Draw program opened directly from the LineDesign Info object in the folder on the left (a simple left mouse button click on the object did it).

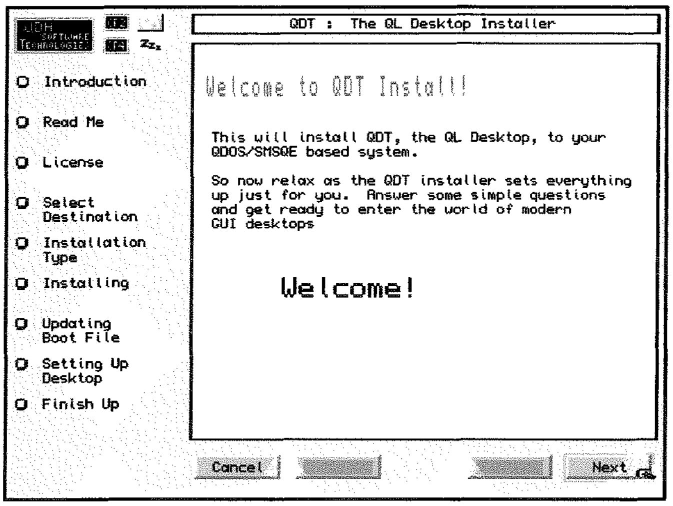
There are two open folders with objects in each of them representing things like text files, actual programs, more folders, executable folders (LineDesign is one and shows an 'X' within the folder - it is actually a folder that, with a single click, runs a file within the folder or opens a file within the folder with another program), and graphics.

The last piece to point out here is a drop down menu for the Applications folder, showing all the options available from within the folder.

This screenshot is a very good sampling of what is to come. But there is a lot more work to do before we go into Beta and then finally ship QDT. I am now expecting Beta to happen shortly after the first of the year and then actual shipment of the first QDT towards the end of the first quarter (hopefully!!!).

Current Activities

My current efforts are focused on a couple of places. The first is the installer (picture is also included on the next page). This installer is a fairly detailed program in itself. It will walk the user through full installation, will check the system and allow boot file updates (configurable), automatic or assisted desktop setup, etc. The intent is to make it so just about anyone can install and configure QDT by simply answering some very basic and hopefully simple questions.



QDT : The QL Desktop Installer

Welcome to QDT Install!

This will install QDT, the QL Desktop, to your QDOS/SMSQE based system.

So now relax as the QDT installer sets everything up just for you. Answer some simple questions and get ready to enter the world of modern GUI desktops

Welcome!

I am also working on converting all the object icon color mapping to an internal 256 color palette that corresponds to a better spread of the systems 16000+ colors and which should map closely into the Aurora's fixed color palette. This will allow me to keep the size of the object icons smaller, give a better spread and ordering of colors that the default SMSQ/E color palette has (for this application), match QDT to all systems including Auroras, and not mess with the system palette itself, just in case someone else is depending on the palette or decides to change it.

The other major activity is to finish the Notebook data saving. Currently notebooks extract the data that they show directly from the QDT database but I have not enabled saving changes back to QDT yet. Once this is done, it is a relatively 'simple' job to add the Add Object capability directly from the desktop, not to mention make it so any changes entered at the desktop level actually are kept.

Once these three activities are done (plus a list of smaller items), I will be ready to start a limited Beta testing run. It should be very interesting to see how all this works on different systems and with different boot and file setups.

And Where Does the Magic Happen

the final image is of my main working space. With the addition of an iMac to my household earlier this year, my dining room table has been completely taken over. I do my main development on my PC laptop seen on the left while I check things on QDT (Virtual PC and QPC emulator), take notes, and do research on my iMac (not to mention play my iTunes, follow the copious amounts of email on the QL mailing list, etc). I have found that, by moving my center of activity from my smaller desk in my upstairs room to downstairs, it is easier to sneak a moment or two to do work on QDT as time allows.

Web Updates:

The last note for everyone here is that I finally am able to get back to updating my web site on a regular basis. I spent the last several months learning and converting to a new web publishing program (on my iMac). Everything has now been converted and you can actually see the images included here in full color and resolution on my sight. Be sure to stop by now and then to see the latest updates and screen captures. My software development main page is

http://www.jdh-stech.com

Just follow the links to the QL and QDT sections.

And as usual, and comments or suggestions are welcome. Please send them directly to me at

jhunkins@jdh-stech.com

3D Perspective Animation - Part 1

Stephen Poole

Way back in 1984 when I bought my first QL, I bought two books which were very stimulating. The first was 'GUIDE PRATIQUE DU SINCLAIR QL' by E.Tenin & J.M. Van Thong. Chapter 5.2.7 discussed graphics and at page 175 began the listing of a program called 'VISUALISATION DE CORPS 3-D', ammounting to 451 well-structured lines of code, whose purpose was to allow the creation, manipulation and visualisation of 3D-objects generated by revolution. However, the routine 'proj_pers', (or perspective_projection), on lines 2300 to 2360 produced a very deformed view and it was clear that this could be done better. The second book was 'EXPLORING ARTIFICIAL INTELLIGENCE ON YOUR SINCLAIR QL' by Tim Hartnell. Chapter seven comprises a program called 'Blockworld', based on Terry Winograd's landmark program 'SHRDLU' wherein a simulated robot obeys commands written in everyday english and writes replies to complex questions such as: 'Will you please stack up both of the red blocks and either a green cube or a pyramid?', 'OK', 'Which cube is sitting on the table?', 'OK, THE LARGE GREEN ONE WHICH SUPPORTS THE RED PYRAMID', etc. Tim concluded by suggesting improvements to the program such as: 'Create your own Blockworld from scratch, in which the blocks and other elements can be moved in three dimensions, so "in front of" and "behind" are valid commands'.

At that time, my full-time job was Surveying farms in need of Land Drainage systems and I was busily juggling with trigonometry on a daily basis. It seemed to me that it would not be too difficult to set up a 3D-routine, and after sitting down and setting-out a technical-drawing of the problem, I defined a series of variables which could then be programmed conveniently. (Unfortunately, these sketches disappeared with many others after our cat tore them to shreds in the bottom of a cupboard, before giving birth to a litter of kittens!).

Taking Tenin & Van Thong as a model, I designed the program to READ its information from DATA statements which would then be displayed in three views. This simplifies input, as you see straight away if you have made any mistakes as the objects appear deformed. Moreover the Data lies conveniently beside the procedures which link the coordinates as LINEs. Thus debugging is easy. As each set of Data for an object has its own display procedure, it is simple to progressively build up the complexity of the display. So first the program calls the series of CREATE and DRAW procedures as it fills up its points-array. Then comes the interesting part, the perspective code. I assumed that one must draw a line from the eye, through a window, which in this case corresponds to the monitor-screen, to a point central to the scene. The coordinates for these two end-points are coded into the program, and serve as a base-line to which all other coordinates in the scene can be related, so that their intersection with the window can be easily computed and plotted. This works well as long as the viewing-point lies at the same altitude as the 'central'-point. (Otherwise, if the viewing angle is sloping, all the coordinates must be stored in an array, and rotational-factors applied to redress their horizontality). So the program calls the routine 'VIEW_angles', which sets up the reference-line for each view. Then for each point the routine 'View_' is called, which works out the polar angles of slope and azimut relative to the reference-line. By this means, the intersection-points can easily be calculated and the new screen-coordinates for each point are stored in an array. Now all that remains is to redraw the whole screenful in perspective, and then move the reference-line to view the scene from a new angle. This whole process is redone for each movement of the reference-line, and frame by frame there is built up a moving animation via the routine 'Trajectory'. This will rotate around the exterior of the scene, giving a view from each corner, then zoom in along the buildings and down under them. At the same time a plane is flown well above the houses. To accelerate the animation, just keep pressing a key. Of course this program, which was written in 1984-5, could be considerably improved. Particularly, it suffered from the weakness of ATAN under QDOS, which I later debugged and which is fixed if you have SMSQ/E. Moreover the graphics edges are slightly warped, due to an oversight on my part, (but this is corrected in the next artical). In addition, it has no sloping views, no hidden faces and no instant animation. But it has the advantage of being simple and as such is

a good introduction for the average reader, and thereby should go a good way to demystifying 3D programs. As this artical should demonstrate, 3D animation is not difficult, but building up complex scenes is tedious as you have to enter enormous amounts of data. Indeed in 1986 I stopped developing 3D-animation-programs because of the time the JS-QL took to produce each screen, which was really too long to make program development practical. Certainly, things are better with a Super Gold Card running under SMSQ/E, but I long for the sort of speed a good Geforce-4 graphics card would give, (ie: millions of facettes per second, allowing real-time output at over 30 frames per second). I would be interested to know what performance can be achieved with the Q60-80. Try experimenting with 'Trajectory', to change the viewing-positions during the animation, noting that the 'central'-point cx,cy is set in the center of the scenery to facilitate output. If you are really adventurous, set cx & cy for each x, y & z so as to do a camera-travelling trajectory, using the image select-count. This first artical uses an early program 'World_bas' because it runs on a basic QL with no need for toolkits. In 1985 this program aroused some interest here in France, and was even filmed! It can produce output similar to that produced by the QL Flight Simulator. As some of you may have noticed, the various 3D routines I printed in Quanta all use basically the same code as this one, (and the next). In my next artical I will describe a fast debugged 3D-Animation program which Jochen will help me to adapt to run on all QL-related machines, as at the moment I don't have the documentation to achieve this.

```
100 ::
110 REMark WORLD_bas, by S.Poole. v1984-5, (v28sept2002).
120 REMark LRUN under QDOS or SMSQ/E.
130 CLEAR: main: AT 22,1: INPUT'Done...>';i$ : CLS: STOP
140 :
150 DEFine PROCedure main
160  WINDOW 512,256,0,0: BORDER 0: CLS: WINDOW 256,206,256,0
170  pse=99: FILL 0
180  REMark Count the number of edges:
190  field=8: house=10: scraper=8: scene=25: plane=6
200  REMark q=1 is 3-views, q=2 is perspective-view.
210  init: q=1: create_scene
220  CLS : q=2: trajectory
230 END DEFine main
240 :
250 DEFine PROCedure CLS_
260 LOCal f
270  PAUSE pse: FOR f=1 TO 3: CLS#f: END FOR f
280 END DEFine CLS_
290 :
300 DEFine PROCedure create_scene
310  CLS_
320  AT#0,0,0: PRINT#0,'field      ': create_field  : draw_field  : CLS_
330  AT#0,0,0: PRINT#0,'house      ': create_house  : draw_house  : CLS_
340  AT#0,0,0: PRINT#0,'sky-scraper': create_scraper: draw_scraper: CLS_
350  AT#0,0,0: PRINT#0,'scenery    ': create_scenery: draw_scenery: CLS_
360  AT#0,0,0: PRINT#0,'aeroplane  ': create_plane  : draw_plane  : CLS_: CLS#0
370 END DEFine create_scene
380 :
390 DEFine PROCedure trajectory
400 REMark nb: Nudge & fudge on exact quadrants!
410  WINDOW 512,256,0,0: CLS: SCALE 500,-500/1.5,-500/2
420  REMark First set the 'Central' point:
430  cx=k+(175): cy=k+(175): ct=8
440  REPeat loop
450     ct=ct+1
460     SELect ct
470       REMark first rotate around the scene:
480       =9 : x=-351: y=350 : z=50
490       =10: x=-351: y=700 : z=50
500       =11: x=1   : y=700 : z=50
510       =12: x=351 : y=700 : z=50
520       =13: x=701 : y=700 : z=50
530       =14: x=701 : y=350 : z=50
540       =15: x=700 : y=1   : z=50
550       =16: x=701 : y=-350: z=50
560       =17: x=1   : y=-350: z=50
```

```
570       =18: x=-351: y=-350: z=50
580       =19: x=-350: y=1   :  z=50
590       REMark then zoom in behind the plane:
600       =20: x=-501: y=110 : z=120
610       =21: x=-401: y=50  : z=110
620       =22: x=-300: y=1   : z=100
630       =23: x=-200: y=-41 : z=90
640       =24: x=-70 : y=-101: z=80
650       =25: x=-50 : y=-161: z=70
660       =26: x=-40 : y=-171: z=60
670       =27: x=-30 : y=-181: z=50
680       =28: x=-20 : y=-191: z=40
690       =29: x=-10 : y=-201: z=10
700       =30: x=-1  : y=-211: z=-21
710       =31: x=-1:   y=-211: z=-50
720       =REMAINDER : EXIT loop
730    END SELect
740    cz=k+z: REMark Make 'central' altitude equal to eye height.
750    draw: fly_plane
760  END REPeat loop
770 END DEFine trajectory
780 :
790 DEFine PROCedure draw
800  INK 7    : VIEW_ x,y,z
810  INK 2,6,2: draw_scenery
820  INK 4,0,2: draw_field
830  INK 7,4,2: draw_house
840  INK 7,4,2: draw_scraper
850  INK 7,4,2: draw_plane
860  BEEP 123,45: PAUSE pse
870 END DEFine draw
880 :
890 DEFine PROCedure view_angles
900   CLS: vx=k+x: vy=k+y: vz=k+z
910   xx=vx-cx: yy=vy-cy: zz=vz-cz: rr=1
920  REMark Calculate slope & bearing:
930   c=0: IF xx: c=ATAN(yy/xx): END IF : r2=((xx^2)+(yy^2))^.5
940   b=0: IF r2: b=ATAN(zz/r2): END IF :  r=((r2^2)+(zz^2))^.5
950 END DEFine view_angles
960 :
970 DEFine PROCedure VIEW_(x,y,z)
980  view_angles: INK 7
990  FOR f=1 TO pts
1000       REMark Get 3-view x,y,z:
1010      px=Tb(f,1): py=Tb(f,2): pz=Tb(f,3)
1020      lx=vx-px: ly=vy-py: lz=vz-pz
1030      lh=((lx^2)+(ly^2))^.5
1040      Lr=((lh*lh)+(lz*lz))^.5
1050      REMark Calculate slope & bearing:
1060       d=0: IF lh: d=ATAN(lz/lh): END IF : e=d-b
1070       h=0: IF lx: g=ATAN(ly/lx): END IF : h=g-c
1080      REMark Calculate perspective points:
1090      REMark (see next artical to remove warping).
1100       xx=r*TAN(h)*-1: yy=r*TAN(e)*-1
1110       Pt(f,1)=xx: Pt(f,2)=yy
1120       POINT xx,yy
1130  END FOR f
1140 END DEFine VIEW_
1150 :
1160 DEFine PROCedure init
1170   pts= field+house+scraper+scene+plane
1180   k=1000: REMark Add k to avoid negative values.
1190  REMark Tb=3 views, Pt=perspective coordinates,
1200   DIM Tb(pts,4), Pt(pts,4)
1210   WINDOW 512,256,0,0: CLS: WINDOW 256,206,256,0
1220   SCALE 100,-50,-50
1230   OPEN#3,scr_256X99a0X0: SCALE#3,50,0,0
1240   FOR f=0 TO 3: BORDER#f,0: PAPER#f,0: INK#f,7: CLS#f
1250   CSIZE#3,3,1: AT#3,1,1: PRINT#3,'3D PERSPECTIVES'
1260 END DEFine init
1270 :
1280 DEFine PROCedure create_scenery
```

```
1290  SCALE#1,k,k-300,k-300
1300  SCALE#2,k,k-300,k-100
1310  SCALE#3,k,k-300,k-300
1320  REMark Set grid coordinates:
1330  w0=0: w1=200: w2=250: w3=300: w4=350
1340  RESTORE 1400: ff= field+house+scraper
1350  FOR f=1 TO scene
1360      FOR j=1 TO 3
1370          READ rd: rd=rd+k: Tb(f+ff,j)=rd
1380      END FOR j
1390  END FOR f
1400  DATA w0,w0,10, w0,w1,10
1410  DATA w0,w2,10, w0,w3,10
1420  DATA w0,w4,10, w1,w4,10
1430  DATA w2,w4,10, w3,w4,10
1440  DATA w4,w4,10, w4,w3,10
1450  DATA w4,w2,10, w4,w1,10
1460  DATA w4,w0,10, w3,w0,10
1470  DATA w2,w0,10, w1,w0,10
1480  DATA w1,w1,10, w1,w2,30
1490  DATA w1,w3,50, w2,w3,80
1500  DATA w3,w3,110, w3,w2,140
1510  DATA w3,w1,170, w2,w1,200
1520  DATA w2,w2,270
1530 END DEFine create_scenery
1540 :
1550 DEFine PROCedure draw_scenery
1560  REMark draw lines by point-numbers:
1570  INK 2,0,3: FILL 1
1580  FOR ls=27 TO 42: L ls,ls+1: END FOR ls
1590  :
1600  L 42,27: FILL 0: INK 2
1610  FOR ls=42 TO 50: L ls,ls+1: END FOR ls
1620  :
1630  L 27,42: L 28,43: L 29,44: L 30,45
1640  L 32,45: L 33,46: L 34,47: L 36,47
1650  L 37,48: L 38,49: L 40,49: L 41,50
1660  L 43,50: L 44,51: L 46,51: L 48,51
1670 END DEFine draw_scenery
1680 :
1690 DEFine PROCedure create_scraper
1700  SCALE#1,340,k-70 ,k-10
1710  SCALE#2,170,k+100,k-10
1720  SCALE#3,170,k-10 ,k-10
1730  RESTORE 1800
1740  ff= field + house
1750  FOR f=1 TO scraper
1760      FOR j=1 TO 3
1770          READ rd: rd=rd+k: Tb(f+ff,j)=rd
1780      END FOR j
1790  END FOR f
1800  DATA 120,35,10, 120,60,10, 160,60,10
1810  DATA 160,35,10, 120,35,60, 120,60,60
1820  DATA 160,60,60, 160,35,60
1830 END DEFine create_scraper
1840 :
1850 DEFine PROCedure draw_scraper
1860  L 19,20: L 20,21: L 21,22: L 22,19
1870  L 23,24: L 24,25: L 25,26: L 26,23
1880  L 19,23: L 20,24: L 21,25: L 22,26
1890 END DEFine draw_scraper
1900 :
1910 DEFine PROCedure create_house
1920  SCALE#1,200,k-60,k
1930  SCALE#2,100,k,k
1940  SCALE#3,100,k,k
1950  RESTORE 2020
1960  ff=field
1970  FOR f=1 TO house
1980      FOR j=1 TO 3
1990          READ rd: rd=rd+k: Tb(f+ff,j)=rd
2000      END FOR j
2010  END FOR f
2020  DATA 30,30,10, 30,50,10, 50,50,10
2030  DATA 50,30,10, 30,30,20, 30,50,20
2040  DATA 50,50,20, 50,30,20
2050  DATA 40,30,35, 40,50,35
2060 END DEFine create_house
2070 :
2080 DEFine PROCedure draw_house
2090  L 9,10: L 10,11: L 11,12: L 12,9
2100  L 9,13: L 10,14: L 11,15: L 12,16
2110  L 13,14: L 14,15: L 15,16: L 16,13
2120  L 13,17: L 14,18: L 15,18: L 16,17
2130  L 17,18
2140 END DEFine draw_house
2150 :
2160 DEFine PROCedure create_plane
2170  SCALE#1,300,k-200,k
2180  SCALE#2,150,k-75,k+75
2190  SCALE#3,150,k,k
2200  RESTORE 2270
2210  ff=scene+field+house+scraper
2220  FOR f=1 TO plane
2230      FOR j=1 TO 3
2240          READ rd: rd=rd+k: Tb(f+ff,j)=rd
2250      END FOR j
2260  END FOR f
2270  DATA 35,225,110, 15,222,95
2280  DATA 15,227,95, 1,224,100
2290  DATA 1,190,97, 1,260,97
2300 END DEFine create_plane
2310 :
2320 DEFine PROCedure draw_plane
2330 INK 7
2340  L 55,52: L 52,53: L 53,54: L 54,52
2350  L 52,56: L 56,53: L 53,55
2360  L 55,54: L 54,57: L 57,52
2370 END DEFine draw_plane
2380 :
2390 DEFine PROCedure create_field
2400  SCALE#1,400,k-130,k
2410  SCALE#2,200,k,k
2420  SCALE#3,200,k,k
2430  RESTORE 2490
2440  FOR f=1 TO field
2450      FOR j=1 TO 3
2460          READ rd: rd=rd+k: Tb(f,j)=rd
2470      END FOR j
2480  END FOR f
2490  DATA 20,20,10,  20,70,10,  170,70,10
2500  DATA 170,20,10, 170,20,17,  20,20,17
2510  DATA 20,70,17,  170,70,17
2520 END DEFine create_field
2530 :
2540 DEFine PROCedure draw_field
2550  FILL 1
2560  L 1,2: L 2,3: L 3,4: L 4,1
2570  FILL 0
2580  :
2590  L 5,6: L 6,7: L 7,8: L 8,5
2600  L 4,5: L 1,6: L 2,7: L 3,8
2610 END DEFine draw_field
2620 :
2630 DEFine PROCedure L(a,z)
2640  REMark q=1 is 3 side-views:
2650  IF q=1 THEN
2660     x1=Tb(a,1): y1=Tb(a,2): z1=Tb(a,3)
2670     x2=Tb(z,1): y2=Tb(z,2): z2=Tb(z,3)
2680  END IF
2690  REMark q=2 is Perspective-view:
2700  IF q=2 THEN
2710     x1=Pt(a,1): y1=Pt(a,2)
2720     x2=Pt(z,1): y2=Pt(z,2)
```

```
2730  END IF
2740    LINE#1,x1,y1 TO x2,y2
2750  IF q=2: RETurn : END IF
2760    LINE#2,x1,z1 TO x2,z2
2770    LINE#3,y1,z1 TO y2,z2
2780 END DEFine L
2790 :
2800 DEFine PROCedure fly_plane
2810  REMark Just in a straight line:
2820  st=ff+1: pl_ed= st+plane-1
2830  FOR fp= st TO pl_ed
2840      Tb(fp,1)= Tb(fp,1)+50
2850      Tb(fp,2)= Tb(fp,2)-25
2860      Tb(fp,3)= Tb(fp,3)+5
2870  END FOR fp
2880 END DEFine fly_plane
2890 ::
```

Galloping to Christmas by way of Middle Earth

Tony Firshman

After the recent unfortunate, sometimes uninformed, and sometimes downright silly emails in ql-users mailing list on the internet about the SMSQ/E license, I was absolutely delighted to get a letter from an 'octogenarian' customer (one of the very few customers of any age this year) recently to remind me what I like about the QL scene, and why I am still around.

I repaired her QL (for the second time after a ten year gap), but forgot to tell her in advance that replacement mdv hardware was extra. I simply scribbled a note on the invoice saying something like that, and that it was her lucky day as she got it free.

Incidentally the new membrane I fitted in 1990 was still in working order, though brittle. It supported my theory that the main reason membranes fail inside closed QLs is that the tails were bent very hard back on themselves by Thorn-EMI just as they emerge from under the metal plate. Give the tails a gentle curve and the membrane will last - even after it becomes brittle.

Here is her reply:

"Dear Sir,
Forgive the formality. I don't know your name.
I think you really should know just how much your cryptic message on my invoice (No. S07585) encouraged me. It came at a time when other things were not going well and my spirits were low. It was not just the sparing of my bank account, very welcome with Christmas galloping up, but the integrity and generosity translated into action and the humerous [sic] explanation. As an octogenarian I deeply appreciate that. Thank you very much.
It is great to have my QL back in service.
Yours sincerely,"

The sequel to this was her arriving at the recent London show by public transport to thank me in person.

She left with Xmas gifts of a double disk drive (courtesy of the London group - thanks Malcolm) and a Trump Card that was gathering dust in my storeroom. After 15 years of successful microdrive use (who else can say that?), she will be, I hope, suitably gobsmacked.

I hope she is now enjoying the pre-Xmas gallop.

.... and that London show.

I arrived at around 9am at the Welsh Congregational Chapel in Southwark. Everything looked shut, but the side door was open. Sounds of lots of voices. "That is odd", I thought. I am usually the first to arrive. Open the door, to be greeted by people in strange outfits, some with very odd whiskers. People on the floor in sleeping bags. People half dressed. People eating breakfast. Swords. Short hairy legs. Very odd headgear. Foreign tongues. Red striped trousers. Something that looked like a breastplate. Lots of noise and hilarity.

Am I in the wrong chapel - wrong London? It is all beginning to sound the start of a Tolkien trilogy.

"Did you know that we are having a computer meeting here?" "It is OK, we will be out by 10:30!" was the reply.

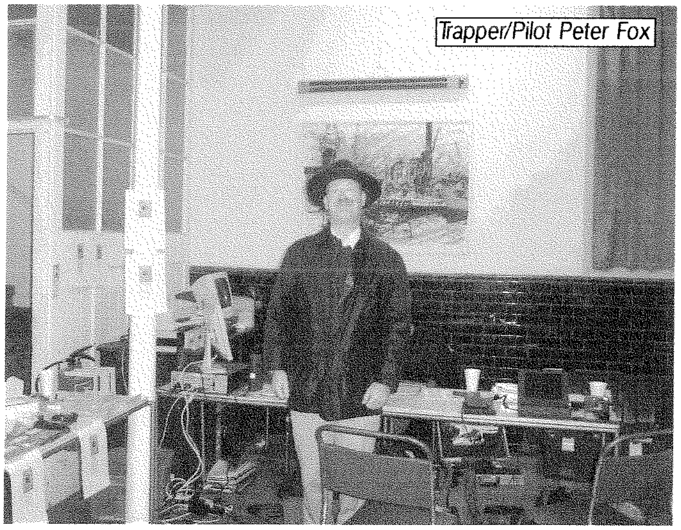
I found the Scottish caretaker - yes he did think 20 years ago when he took the job that this was a mite ironic. "It was all OK when I found out we both hate the English". "That is OK - I am half Welsh" I said. He said they were actually meant to get out by 10:00. We (yes a few more had arrived by then) gently pointed out (while laughing) that we needed to set-up. A compromise of 9:30 was agreed.

It turned out later that they were sword dancers from "Stevenage Sword" attending the annual "Thrales Rapper Weekend of Dance". From the schedule, I found there was dancing mentioned, but pubs got a bigger word count. Well is that any more strange than rituals around little black boxes?

During my setting up, someone arrived wanting his QL mended. After a bit of toing and froing he eventually managed to say he was in a bit of a rush. Switched on my Microvitek without plugging it into the QL. CRACK. It was very humid in the basement, and it was suffering from condensation. I said to the guy - "Sorry, I have to repair my monitor first!". I found a portable gas heater, and that worked a treat. I had never really seen arcing in a TV before. The spark was a good 4 inches long at least - amazing. No wonder 8301s get zapped by Microviteks, which are renowned for arcing, even when dry.

At last after 30 minutes I was able to test his QL. It was working 100%, even the membrane. "What was worng with it?" I asked him. "I can't remember - it was too long ago"

I pointed out the aged note stuck to it - which I couldn't really read. "Works OK, but doesn't start up with the interface". Hrmm. Mind you that can happen when expansion pins are bent - but J1 was OK. He is using the interface now on another QL, so I guess he didn't plug it in fully. Damn - forgot to ask him what power supply he used.


Trapper/Pilot Peter Fox

Later on someone had a dead power supply - tightened the mains plug screws, and it worked again. It was a day of repairing working hardware. It is quite amazing how the same issues recur - I first pointed out mains plug problems in 1986. As connectors can get quite warm in use, the continual expansion/contraction can very easily make screws work loose. In my mains products, I seal connections with hot glue (my cure-all!).

Keith Mitchell and I enjoyed a wrestle with his laptop. His wife suffers from premature line drop (no laughter please). I suggested installing a modem Roy had there, to show him the relevant DUN setting. Did you know that USR have two power supplies - 18V and 9V. The modems look identical, and there is no clue as to which voltage is needed - amazing. "I have no serial lead" said Roy. I went to ask Darren if he had one, and he said "There is one on Roy's table". We both died laughing. Problem - we needed one end to be 9D. Roy had this natty gender changer - about 5 way in any combination. No good though as Keith's power supply connector was in the way. We then piggybacked it off two std in-line ones and it worked!

I asked Bill Richardson if he had found the person who wanted to sell him some QLs "Yes - that is the only reason I have come". Bill is another person that day who proved that octogenarians are forever young.

It was actually a pretty lively show for QL matters, and was memorable for a presentation to those two ex-Quanta stalwarts, John Taylor and Bill Newell. It was nice to see them there as civilians for a change.

I sold what seemed like hundreds of s/h microdrive cartridges - amazing.

.... and Malcolm Cadman's rolls, with complimentary trimmings and organic pickles, were so good and so cheap I never got around to the lunch I brought with me.

I hurtled off promptly to Evensong and a subsequent Bach/Handel rehearsal - phew what a day.

Well done again to the organisers - but avoid the Rapper weekend next time.

Fun Stuff

Per Witte writes with a couple of humorous QL-related stories:

An acquaintance called me the other day saying his computer, an ageing original QL, is broken. "What's the problem?" I queried. "It doesn't work", says he, helpfully. "I'd like you to come round and take a look." "How about we do a little trouble shooting first?", says I. "Trouble what?" quoth he. "Trouble shooting", I repeat. "You see, before I come over I want to be quite sure that the machine really doesn't work." "Alright, then", says he, "'ang on". He lays the receiver down and I hear the sounds of rummaging at the other end. I wonder what the hell he's up to. Then I hear a click followed by a loud BANG! and in the subsequent silence I hear the gentle patter of falling bits of plastic. The acquaintance picks up the receiver again and says "Ok. Done that. What next?"...

One of our great QL-luminaries was once out walking in the woods, when he espied a small brass lamp lying in the undergrowth. He picks it up and rubs the dirt off with his sleeve. Somewhat to his surprise a genie pops out, bowing obsequiously: "Oh master, it is in my power to grant you a wish. Speak! Thy whim is my law!" "Hm" quoth the said luminary, "I thought it was customary in such cases to be granted three wishes, not one." "Alas! It is the times, lord. What with 9-11, the current interest rates, the NASDAQ going through the floor, globalisation, the Microsoft monopoly and what have you, even us spirits have to tighten our belts. One wish it is. Now what will it be, lord and master?" Our luminary ponders deeply a moment then, bringing out his pocket diary, points to a map and says "See, this is the Middle East. I want you to stop these guys from fighting and killing each other. I want you to make peace in the Middle East!" "Good gracious me, my lord" says genie aghast. "These people have been at war for decades – centuries! Nay, come to think of it, millennia! Impossible! Think of something else, I beg you. How about a shiny new Mercedes 500 SLK? Or a super-duper new PC..?" Our luminary gives him a withering eye, but then a happy thought strikes him: "I wish", he says, "that the QL-world would smooth over their differences, put aside their idiosyncrasies and extreme individualism – for a little while, at least – and get together to produce an intelligent, functional and backwardly compatible file system, allowing for unlimited directory depth, long file names, and a sensible solution to the underscore/backslash problem that we can all agree on. Yes! That is my wish!" The spirit is silent a moment and then, with a deep sigh of resignation, says "Lets have a look at that friggin' map again, eh."

Pirates and mad Sheep

Tony Tebby

The recent discussion on the newsgroup and the problems related to D&D not being an SMSQ/E reseller made Tony write an interesting mail - we re-print most of it with his permission as it will be very interesting to QLers who have no access to the newsgroup.

It is not just about the reseller trouble (we have removed the personal bit, as we hope the situation between D&D and the registrar is sorted out by the time you read this, it also explains the license and adds some comments about the trouble around the license discussion several months ago.

While walking on the fells, I came across a sheep with its horns entangled in a wire fence. With great difficulty I managed to free it and, by way of reward for saving it from a certain, lingering death, it tried to take my fingers off.

More about mad sheep and fingers later. This note is written in response to some very strange comments that have been published in this list and elsewhere.

Before attempting to set the record straight, I shall try to explain the principle of royalties (or licence fees).

The principle is that an author devotes a few hours, a few months or many years creating what the law considers, rather quaintly, to be a "work of art". A "work of art" is not like other commodities. If you buy a book, the author gets a royalty on the sale, but, if you do not like the story, or if there are fatal flaws in the story line or even grammatical errors, you cannot get your money back.

If an author is lucky or has a good agent (being a good author has almost nothing to do with it) his book becomes a best seller and the author pockets royalties out of all proportion to the effort that went into the book. Usually, the author (or the painter, the composer etc.) is unlucky.

In all cases, however, the royalties only come some time after the work has been done. The royalties are a recognition of the work that has already been done. An author does not receive royalties to pay for new "works of art" or to improve existing works of art. Royalties are a payment for work that has already been done and nothing else.

Royalties are due by anyone who makes a copy of a "work of art". A licence is slightly different - it is the permission to copy a "work of art". Making a copy without a licence is a criminal act. A licence may be tied to a support contract, but in the case of SMSQ-E, the "licence fee" has always been pure author's royalties (legally and fiscally).

I wrote QDOS for the QL, it was not perfect, but it sort of worked. It was not the operating system that I would have liked to write, but it was the operation system I was paid to write. After the demise of Sinclair, I was under considerable pressure to provide a legal, maintainable alternative to pirated copies of QDOS (there are still pirated copies of QDOS being sold 16 years later). No one was prepared to pay me to do it, but I gave in and did it anyway.

If you take all the royalties I have received for SMSQ-E and multiply by 10, it would still not pay for the development that was done for the various machines SMSQ-E was made available on. The only payments that I have received for support have been from a small number of generous people or groups who have contributed to the development of specific improvements that were made available to everyone. I never really thought releasing SMSQ-E would be worthwhile, but I was naïve enough to think that it might save my fingers. It didn't.

Now for setting the record straight.

Wolfgang Lenerz

About a year ago, suggestions started to be made seriously to make it possible for development to continue by making the SMSQ-E source publicly available. Nothing particularly radical about that, authors of books do it all the time, and I had already communicated complete or partial sources to various people who had requested them.

Wolfgang Lenerz consulted me before setting off to Eindhoven to discuss the proposals with "interested parties". I do love to say "I told you so". I told him that if he went to Eindhoven, he would be "voted" to run the whole show, and what this was likely to do to his life. But he went anyway.

Wolfgang Lenerz has been working (unpaid) to try to discourage aggressive lockout policies designed try and capture a larger share of the QL "market" at the expense of QL users. I.e. he has been trying to maintain a coherent cross platform environment. It is possible that you may have different ideas on how this can be achieved, but Wolfgang Lenerz has no personal or commercial

CAMBRIDGE Z88

OFFICE/FAX 01494-871319 (EEC) *W.N. Richardson & Co.*

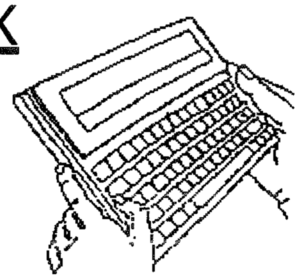
MOBILE: 07808 576118
E-MAIL: WNR@COMPUSERVE.COM

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL, PC AND MAC.

THE CAMBRIDGE Z88 A4 NOTEBOOK

WITH BUILT-IN WORD PROCESSOR, SPREADSHEET, DATABASE, BASIC, CALCULATOR, CLOCK, ALARM, CALENDAR & VT52 TERMINAL.
USES 4XAA ALKALINE CELLS (ca. 20 HOURS) ✱



WE DO REPAIRS, PART EXCHANGES, AND BUY Z88's & PARTS

NEW! CAMBRIDGE Z88 WITH OZ4+512k Internal Ram £150

ALSO AVAILABLE, 512k RAMPACK £40

DESCRIPTION	UK	USA
CAMBRIDGE Z88 ISSUE4-128K	**£120**	**$180**
CAMBRIDGE Z88 COMPUTER	**£99**	**$130**
☆**RECONDITIONED Z88**	**£60**	**$90**
32k RAMPACK	£18	$25
128k RAMPACK	£38	$50
32k EPROM PACK	£18	$25
128k EPROM PACK	£24	$33
256k EPROM PACK	£55	$75
EPROM ERASER	£32	$43
PARALLEL PRINTER LEAD	£39	$50
SERIAL PRINTER LEAD	£12	$16
✱ **MAINS ADAPTOR (230vac; 6v,500ma)**	**£10**	
TOPPER (PROTECTIVE COVER)	£12	$16
CARRYING CASE (PLASTIC)	£10	$12

1/2 PRICE STARTER PACK
1 - Z88 ☆ PLUS
TRANSFER KIT
+ 32K RAM
+ 32K EPROM
£72 ONLY!!

FILE TRANSFER TO OTHER COMPUTERS

	UK	USA
SPECIAL Z88-QL SERIAL LEAD	£10	$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	£2	$3
PCLINK KIT (For PCs)	£25	$35
Z88 TO MAC KIT	£25	$35
Z88 TO BBC KIT	£25	$35

} ALL WITH INSTRUCTION BOOKS, CABLES AND EPROMS AS REQUIRED!

POSTAGE UK UP TO £5. EEC £15. USA £20. OTHER COUNTRIES £30.

ALL THE STOCK IS WARRANTIED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING ARGREED, BUT THE ITEM BEING OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

QL & PC COMPUTER USERS WILL FIND THE CAMBRIDGE Z88 ESPECIALLY USEFUL FOR WORK AWAY FROM THE DESKTOP, WITH TRANSFER PROGS DATA CAN BE SAFELY EXCHANGED WITH THEIR DESKTOP SYSTEM.

W.N.RICHARDSON & CO CONTINUES TO PROVIDE FULL SPARES AND SERVICES FOR SINCLAIR COMPUTERS, QL & THE CAMBRIDGE Z88

stake - he has been working for the benefit of QL users and anyone who says otherwise is LYING.
I suggest you think very carefully about the mentality of those who have thrown the insults that Wolfgang Lenerz has been receiving.

The 10 Euro ($10) royalty
In Wolfgang Lenerz's message in the list he states "Under this licence, only appointed resellers may sell the software, provided, notably, that a 10 euro payment is made to Tony Tebby for each copy sold."
I did not actually negotiate this but it was offered and I agreed. But think very carefully about it. This is a royalty as payment for the original development. It is not a payment for services, further developments etc. Before this arrangement came into force, Jochen Merz collected licence fees as my agent. He kept a share for providing support and passed on the rest as royalties (but his share was not really large enough to pay the time that he spent supporting SMSQ-E - for that he would have needed to take several 100% and I can't afford that).
With the new arrangement, Jochen Merz apparently still collects some licence fees, but he gets no cut at all. This dramatically reduces the licence fees to be paid on "legal" copies of SMSQ-E. This is, apparently not good enough for some people who just do not want to pay at all.

[D&D and Q60 bits removed - see explanation above.]

More in sorrow

I accept the blame for writing QDOS and the consequences, but what have Wolgang Lenerz, Jochen Merz and others done to merit the treatment they are getting - they deserve to keep their fingers.

... and another interesting mail followed:

There have been a number of confusing statements about licences, core software and patches for the Q60. I do not know whether they confused you, they certainly confused me.
A licence is not a restriction - it is a permission. The SMSQ/E licence gives explicit permission to software developers to do a lot of things that they could not do before. The only effect on "end-users" should be more, better quality, support (and still free). If you are a software developer, all you have to do is ask.
SMSQ/E is delivered as a number of modules. The difference between SMSQ/E modules and Windows modules, for example, is that Windows modules are selected and linked together (using the registry) for the hardware configuration on installation (about an hour), whereas SMSQ/E modules are selected and linked together using "soft" linkage blocks in a few seconds every time you boot the system. Admittedly, Windows is more complicated - if you were to include all the core functionality of Windows into SMSQ-E you would need to wait a few seconds more for it to boot.
The modular structure of SMSQ/E is historical rather than logical. There are some small modules such as the hardware initialisation module that are completely hardware environment dependent. There are some large modules that are completely hardware independent, such as the SBasic interpreter. Most of the modules, however, have a set of machine independent core routines with hardware dependent routines and data definitions. In principle, only one version of SMSQ/E needs to be delivered which has all the different modules for all machines with only the "right" modules being selected for the configuration. In practice, the number of different versions bundled together is limited.
To change the behaviour of part of the system, all that a developer / supplier needs to do is to replace the module concerned by a new version add a new module to the end of the operating system "package".
These extensions may modify the core functions (subject to licence conditions) add new core functions (subject to licence conditions) add new luxuries (it would be nice for QL users if you complied with the licence conditions).
When QLCF contributed towards the development costs of the 16 bit colour drivers for the Q40, it would have been much simpler and cheaper to have resticted the colour drivers to Q40 colours only. What was actually developed was a frameword for colour capability from 1 bit to 24 bit with drivers specifically produced for monochrome, QL 2 bit and 3 bit colours and any 15/16 bit fixed colour scheme. The sources were then passed to developers for the Aurora and QPC to develop drivers specifically for their systems.
The intention was to ensure that modified core functions and new core functions remained as compatible as possible with older versions and across the new versions. This is the purpose of the licence. All SMSQ/E developers are building on the back of work done by others. If they make a

significant contribution to the core functions, they may be justified in asking for royalties for their contribution, but that does not justify their refusing to share their work with other developers - the principle is open source not "everyone for himself". The licence conditions are designed to favour this type of co-operative development.

The effect of the licence conditions can be seen more easily if we sort the sheep from the goats in the patched Q60 version of SMSQ/E story. Someone somewhere is confusing two totally separate issues.

The first issue was a bug in the Q40 hardware initialisation module (not in the SMSQ memory manager (software) as stated in this list, but in the initialisation of the MC68040 MMU (Memory Management Unit - hardware)). This was a one bit error in a 1046 byte SMSQ module. A genuine bug which meant that some Q40s (depending on the brand of memory module fitted) sometimes ran more slowly than usual because the caches were disabled. This never happened on my Q40 and it was the sort of error where you could look at the code a hundred times and not spot it.

This affected only the Q40 specific version of SMSQ/E and so there was no impact on other machines. It was fixed in Version 2.99, but the modification was provisionally made to the V2.98 shipped by Peter Graf.

The second issue was the reduced instruction set for the MC68060. The MC68060 cannot execute all the MC680x0 instruction set and if you try to run MC680x0 programs on the MC68060, they quite often crash terminally (reset button time).

For most modules, replacing the deleted instructions by alternative code sequences has no real impact on performance on any machine, so this was done. However, a critical instruction in the QL colour mode drivers is no longer supported by the MC68060.

The general solution is to provide an "illegal instruction" trap to process the exceptions generated when the 68060 came across an instruction that it did not recognise. This is rather slow, but it ensures that the Q60 can also execute any program that uses the full MC680x0 instruction set.

A module incorporating this solution can be added-on, does not require any modification to SMSQ/E and provides a slower QL colour mode than the ideal, but still a lot faster than other machines.

The specific solution is to rewrite the QL colour mode driver to eliminate all "MOVEP" instructions. I had started this work before passing the partially modified MC68060 routines over to Peter Graf. The modified code works on earlier processors, but is not as fast. It is, therefore, necessary to provide separate MC68060 and MC680x0 versions of the GD2 QL mode drivers, even though they are >99% the same.

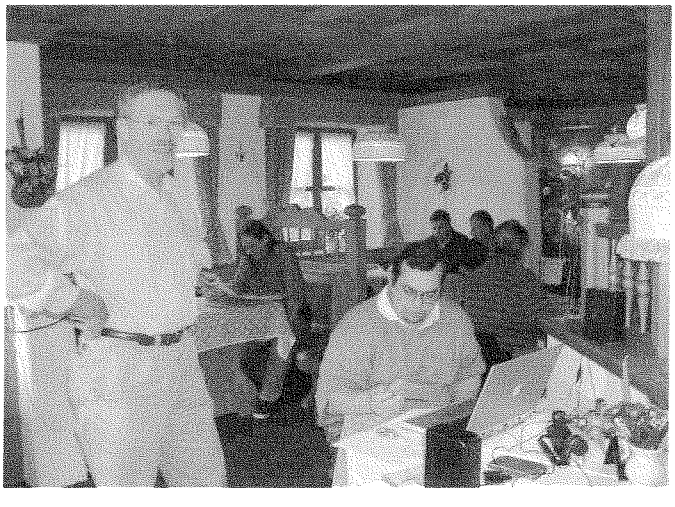
One of the registrar's jobs is to ensure that the modifications are properly recorded. In this case, he would have to ensure that the lower performance MC68060 version did not crawl through into the "standard" versions. Naturally, other suppliers of MC68060 based machine could use this version because it is in the "open source". The Q60 developers would, however, be quite justified in asking for 1% of the price they paid for the Q40 version of the GD2 QL driver - 1% of nothing is not too much to ask for, is it?

QL Show in Berchtesgaden - Germany

Peter Fox

Steve Reyal, Roy Wood and I flew to Salzburg on Friday 4th October in my aeroplane taking around 3 hours for the trip. The weather on the trip was super and we had a good view of much of the countryside.

At Salzburg we hired a car and drove to the Hotel Schwabenwirt in Berchtesgaden to find that our accomodation and the show had been moved to the Hotel Tauernhof which was on the outskirts of the town. The hotel grounds were bounded by a brook and a river in a wooded area which is very attractive.



We had Dinner in the Hotel and after a good night's sleep and breakfast, started setting up for the show in the Restaurant which meant everyone was running late. Once everyone had settled down there were a dozen laptops running QPC2 scattered around the room. There is no doubt that the hotel is an excellent venue although I personally have no experience of the Hotel Schwabenwirt. It is clear that Technology was moving on since information was being transfered between laptops using Compact Flash Media.

We had visitors from Germany who also spent some holidays in this nice area, like they did the year before. Also, QLers from Austria and Italy visited the show.

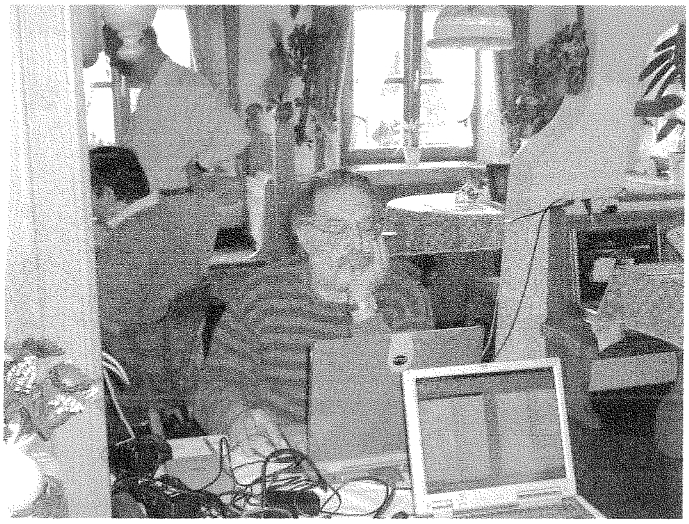
Jochen Merz and Roy Wood were the only dealers present this time - Jochen carried his whole range of software, CDs etc. but, unfortunately, had to carry most of it back home due to the fairly small number of potential customers. Roy was well prepared for this situation.

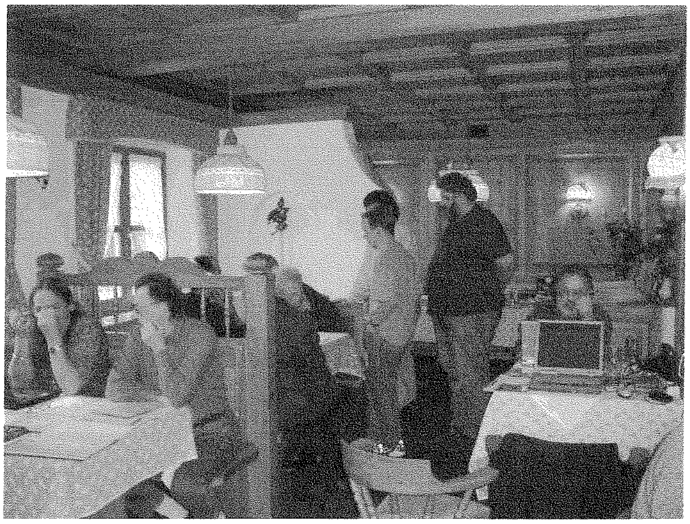
Marcel drove over from Stuttgart for the afternoon taking about five hours and talked to everyone.

There was dinner after the show also at the Hotel which again was perfectly good and enjoyable with a great deal of discussion in two groups, one in German and the other in English.

Everybody had a good time, and we all would like to thank Friedemann Oertel for organising the show and making sure it worked out nicely. We all hope that another show next year will happen again, and look forward to more visitors then. And - Friedemann - make sure the weather will be warmer again, like it was last year!

[Jochen adds: we have already spoken to Friedemann, and it seems that another show will be possible 1st weekend of October 2003 - why not put it into you Agenda NOW?]





D&D's Q60 Offer

Dennis Smith

The Q60 has been available for about a year now and during that time we have made steady consistent sales. After our summer break D & D Systems must return to the Q60 production line. We have looked at what the public tend to want the most and come up with a nice package as listed below:

Midi Tower case – 3 x 5.25" & 2 x 3.5" bays
Q60 @ 60MHz
64MB RAM
CD-ROM 56x
3.5 Floppy disk drive
20GB Hard disk
I/O card, 2 serial ports @ 115,200, 1 parallel port, 1 games port

Keyboard
3 button mouse
65536 Colours (Photo quality can be displayed on the screen)
Stereo sound output
SMSQ/E & QDOS Classic operating systems

This is a 'Plug in and go' system, no messing about. There are a few extras you could have such as a 17inch monitor, Ethernet card or more serial ports. The stereo speakers we supply are plenty good enough for personal use and are based on a 3 speaker system, that is; 1 left, 1 right and 1 sub-woofer (Bass). The speakers are not included in the above package as some people may want to produce the ultimate QL Jukebox and blow the neighbours wall down, in that case get your own big speakers and plug them into the stereo jack. You could use the small internal speaker in the case if you wanted as a low power alternative. Is 64MB RAM not enough? OK, try another 64MB to go with it, total 128MB, lovely. At the time of writing SMSQ/E does use this in a fashion and the coding should be in use as you read this. At the moment I cannot see 128MB being of any use to SMSQ/E but Linux would use it. We have a new version of 68k Linux called Shoestring Linux. This is improved over the first version and has a nice email program on it. The Q60 will browse the Internet, send & receive email using Linux, later on SMSQ/E on the Q60 will be doing this as well. Shoestring Linux does run on the old Q40, slooowly. The keyboard supplied is an English (UK) version, however, you may use your own country specific keyboard and we will deduct the price of this from the offer price.

The price for all the above is £545 all inclusive with only the variable delivery charge to add. This has got to be a good deal for owning the Worlds fastest new QL (and the Worlds fastest new 68k Linux computer).

We are contracted to supply the Q60 as a fully working unit so that means it must be sold with an operating system. The supplied O/S is SMSQ/E although we could have chosen QDOS Classic, SMSQ/E is more developed and takes advantage of the Q60's capabilities better. QDOS Classic is a valuable tool for running that old duff software that SMS does not like, although the version of SMS we use does cope better with old software than the QL did using SMS. Maybe 98% of QL software runs on the SMS Q60 so we are not talking about a problem of any magnitude. Any problems, run QDOS Classic instead of SMS. There is no reason why QDOS Classic cannot be developed far beyond SMSQ/E if the coders were so inclined to do so. By now you should be getting the inclination things are moving forward and development is growing at an increasing rate, this is of course what has been lacking for years in the QL community. The more Q60s sold the faster this development rate will be so I am asking you to stand up and be counted, place an order for a Q60. If we are going to develop this then lets get on and do it.

Q60 production at D&D is not a constant flow because of the relatively low volume and high(ish) stock levels we must maintain. We predict a series production run, get the bits and make that run. Of course we don't make every type of Q60 on a series run and someone will always ask for what we are not making, this is bound to happen. In such a case, which is beyond our immediate control, a delay of four weeks might occur. The Q60 @60 MHz is now our standard, the 66 MHz and 80 MHz versions are available as special orders which means they may be in stock or built in the next batch.

You may have noticed I have not mentioned the Q40 yet so here we go. D&D Systems have never made a Q40 therefore we have never sold one either. We have made a Q40i, which is the same design as a Q40 except it has a RAM capability of 128MB. If the advertising in QL Today is the latest version you should be able to detect an absence of the Q40i for sale because it has been discontinued. However, a Q40i is available as a special order - for the time being - if you really want one. One thing we can supply for the original QBranch Q40 is a matched pair of RAM sticks giving 32MB in total. QBranch were responsible for the production and sale of the Old Q40 so any Q40 matters should be directed to them.

Photographs can be downloaded from digital cameras and displayed on the screen in photographic quality. These can be quickly rotated on screen and I have heard that someone is working on a photo manipulation program to modify/enhance photos. As I mentioned earlier if we can rig up a nice Jukebox program then we have stereo sound and photos on a QL machine, how does that sound?

CD use is not limited on the Q60, the hardware is capable of reading, writing and re-writing CDs and this can be done via Linux. SMSQ/E can read audio and data CDs, the program for writing to a CD needs finishing. This is quite a good step in

QL related computing, imagine for instance a program called QL Gardener (yes, I know it already exists) but think of the new version on CD with hundreds of photos showing the exact flower along with a detailed description, great. This type of thinking is now quite feasible and is indeed dragging back programmers into the QL scene along with some new ones. The Q60 exists so lets start using it seriously.

The options available with hard disks are quite interesting, a compact flash device can be fitted into a 3.5 inch bay (or convert a 5.25inch bay into a 3.5) connected to the i/o card giving a removable hard disk in the form of a compact flash card. These are not expensive, go up to 512MB and four are the size of a matchbox.

Check the latest advert for the up-to-date prices. I wonder if there are people out there worried about the reliability of the Q60? Fear not, we have run these machines for months continuously, no problem. The engineering is sound. Enjoy QLing again with this new lease of QL life. See you at the Shows.

An Inside Story

Geoff Wicks

This article is about programming, but not programming as we normally think of it. How do you transform a skill that has traditionally relied on the intuitive insights of specialist researchers into a computer program that can be used by novices and amateurs? It was over ten years after conception before the program became a reality.

Ten years ago I was living in the Netherlands, and applied for two jobs shortly after one another in which handwriting analysis formed part of the selection process. Psychologists are very influential in the Netherlands and I was used to job application procedures using psychometric testing, but graphology was new to me. I wanted to find out more about it and bought two books on the subject. These were of limited help. If I attempted to analyse the handwriting of a person I did not know, I had no means of knowing how accurate the analysis was. But if I attempted it for a known person, I could not do it objectively because I was influenced the explanations given in the books.

I needed to put a distance between the books and the actual analysis, and being a QL user, wondered whether the process could be automated. One of the two books usefully suggested a structured approach to handwriting analysis, but many of the tests relied on subjective judgements based on years of studying different handwriting styles. These would be difficult to transfer to the factual and concrete demands of a computer program.

My early attempts came to nothing as other things were consuming my time. I was involved in a series of legal and quasi-legal proceedings after I had blown the whistle on some scandals in a children's home. The QL provided relaxation from the pressures I was under, and my QL work was blossoming in other ways. I wrote a series of articles on desk top publishing for QL World and took the first steps towards starting Just Words! Handwriting analysis was put on a back burner and soon forgotten.

I can no longer remember what caused me to return to handwriting analysis, but in the last ten years Just Words! has given me much experience of designing programs from scratch and perhaps I had more confidence. Nevertheless I knew of no similar program, not even for the PC, and I had few illusions about the difficulty of the task. I did not have the confidence to describe the project in detail, but dropped hints through cryptic adverts in the QL press:

"One of our ideas is so esoteric that, as far as we know, it has never been done before. We don't even know if it's possible, but we'll have a try."

By committing myself in public, I was giving myself a spur to complete the job.

I started the project by re-reading the books and making extensive notes. This may seem an obvious way to begin, but not for me. When I did my probation officer training about 30 years ago, note taking was actively discouraged as a sign of insecurity. We were being trained not take notes but to think open-endedly to tackle the problems we would come across in our daily work. Such is my antipathy to note taking that when I started to research this article I discovered I had thrown away most of the notes I had made.

These early notes helped me extract the essentials out of two, sometimes rambling, books. Certain tests were fundamental in any scheme of handwriting analysis. These included the size and style of letters, writing pressure, slant and spacing, legibility and basic letter shapes. Some would be easy to test objectively, others needed illustrated examples and yet others were highly subjective. A further complication is that each handwriting sample is different. For example, the tests to determine writ-

JOCHEN MERZ SOFTWARE

just in time for

CHRISTMAS

SMSQ/E Prices

Due to changes in the licensing structure we are happy to be able to reduce the prices for all SMSQ/E products!

QPC2 Version 3 with SMSQ/E

New Version with complete manual	EUR 99,90

Upgrade to QPC2 Version 3

From QPC1 (or just QPC - the first QPC!)	EUR 69,90
From QPC2 Version 1 (or just QPC2)	EUR 44,90
From QPC2 Version 2	EUR 20,40

You will receive a new QPC part of the manual. Please return QPC master disk. Please note: QPC1 will not be supported on this disk anymore. If you want to continue to use QPC1, then please make a backup-copy of the QPC master disk before you return it to us.

All other SMSQ/E

for GoldCard/SuperGoldCard or QXL or ATARI price per Version	EUR 49,90

Merry Christmas and a Happy New Year!

TERMS OF PAYMENT

Postage and package **[Germany]** EUR 4,86 (if total value of goods is up to EUR 25,- then only EUR 3,32).
[Europe] £4,50 / EUR 7,41 (if total value of goods is under £15 / EUR 25,56 then only £3 / EUR 4,86).
[Overseas] between US$7.50 (1 item) and US$17.50 (maximum).
Cheques in EURO and International Money Order in EURO (both have to be drawn on German banks).
We accept Internation Reply Coupons as payment (can bo bought at any post office): 1 IRC = EUR 1.02 redeeming value.
Exact prices are given in EUR ... the price in £ and/or US$ are approx. conversion based on the exchange rate at the time of creation of this ad. Prices may be adjusted in case the exchange rate falls considerably out of this range in both directions!
We can charge your credit cards in £'s, US$, EURO or in DM - please state the currency you prefer.
Please do not send any UK bank cheques in £ - our banks have increased the fee for handling them by 600% (no joke!) so we cannot accept them anymore, unless you add £6 for clearing the cheque.
E&OE.

Im stillen Winkel 12   D-47169 Duisburg
Tel. 0203 502011   Fax 0203 502012

VISA   MasterCard

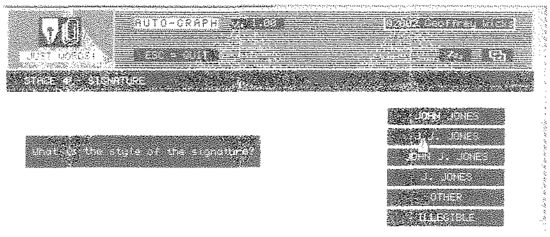
ing pressure can vary according to the thickness of the paper and whether a pen, ballpoint, pencil or felt tip was used. And when all the tests had been completed the results had to be analysed in a meaningful way.

Here another life experience was valuable. About two years ago I earned part of my money by translating tractor surveys for a Market Research company. While there I had been asked to take responsibility for the Dutch and Flemish questionnaires for an international study on eye care. When you have typed, proofread, checked the translation of a 39 page survey, then tested it, discussed it with colleagues from most European countries between the UK, Portugal, Italy and Finland and have done some of the interviewing, you know that survey inside out.

In short I knew how to process complex information using a lengthy questionnaire. Two features are highly important. The wording of the questions must contain no ambiguities and processing of the information is done by numbers. The latter is not surprising because computers work with numbers. In market research these numbers have to be translated into a complex marketing strategy.

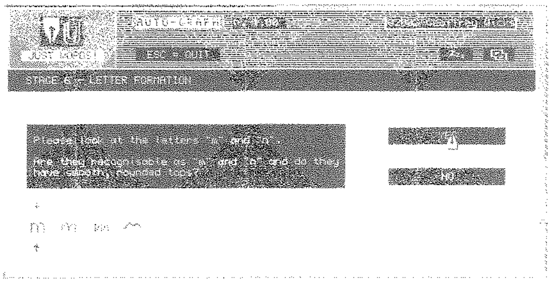
Market research surveys are divided into several sections, the first of which is to gather factual information to determine which questions and sections of the questionnaire are relevant to the person being interviewed. My handwriting questionnaire would begin in a similar way by asking questions about the physical document. Did it have margins? was it a copy or an original? Was it written by pen, ballpoint etc.

The structure of the rest of the questionnaire soon became clear. The second section would be looking at the physical characteristics of the document such as size of margins, the slope of the writing and the relative size of the letters. Another section would have to look at the shapes of individual letters such as "m" or "y". Signatures and addresses on envelopes were worth a section on their own.



What is the style of the signature?

This still left me with the problem of some of the more subjective parts of handwriting analysis. Here again I borrowed from the techniques of market research, namely word association tests. I decided to give users a choice of opposite words to describe the handwriting such as "LEGIBLE - NEUTRAL - ILLEGIBLE" with the instruction that if they had any doubts, then the correct answer would be "NEUTRAL".



Please look at the letters "m" and "n".
Are they recognisable as "m" and "n" and do they have smooth, rounded tops?

The next problem was how to program the questionnaire on the QL. Recent Just Words! programs use the pointer environment and have a distinctive house style, but would this be possible for my program? At the very least I would have to learn new ways of using EasyPtr, especially application sub window menus and sprites. As anyone who has mastered EasyPtr will tell you, this can be a bold undertaking that you do not do lightly. Would it perhaps be easier to go back to non-pointer programming?

This problem was easily solved. Early working versions of the program would use a modified QL character set to display different letter shapes, but I soon discovered that these did not give sufficient quality. Easysprite allowed letter shapes to be drawn in much higher quality. I then tackled the sub window menus. The number and size of menu items would be constantly changing throughout the questionnaire.

Here I used a little deception. If you look at the illustration you will see there appears to be one central window in AUTO-GRAPH. In fact there are two windows. On the left is a normal SuperBasic window and on the right the EasyPtr submenu window. Both have black backgrounds so that they merge into one another. The number and positions of the items in the submenu window may appear to change with each question, but in practice there are always seven items of the same size in the same place.

EasyPtr submenu items are set by the command:

```
MSTAT [#ch%,]{TO} {\}number, stat%
```

Stat% is an array that governs the status of each menu item, that is whether it is available, selected, or unavailable. If stat% = -7 it means that the menu item does not exist, and is therefore not shown on the screen.

Writing the questionnaire was comparatively simple. The questions had to be clear and give simple, unambiguous answers. An important part of handwriting analysis is the relative size of three parts of the letters, the ascenders or upper strokes of letters like "b" and "d"; the body of the letter; and the descenders or bottom parts of letters like "g" and "y". If I had asked a question whether each of these was greater or smaller than one-third of the total letter size, you would be surprised how many people would give contradictory answers such as all three being greater than one third. In practice the program asks only about the ascenders and descenders and then works out what the relative sizes are.

One section of the questionnaire, the word association tests, was a monumental failure. In this section I attempted to determine the general shape of the individual letters by asking if the writing was "smooth", "jagged", "bumpy" or "snake-like". To check accuracy I asked about each aspect twice using different words. The section was too long and confusing, with the result that it was not only of doubtful accuracy, but also dominated and distorted the results of the total analysis. I moved the general shape questions to the individual letter shape section of the questionnaire and specifically asked about the shapes of the letters "m" and "n" where these things could be seen most clearly. The questions that remain in the word association section are far less subjective.

The biggest problem, which probably took up over half of the program development time and over 90% of the testing time, had still to be tackled. How do you turn the answers to the questions into a personality profile?

Many parts of handwriting analysis are concerned with opposites. A writer whose lefthand margins are larger than their righthand margins supposedly have a tendency to live in the past, whereas those whose righthand margins are smaller than their lefthand margins are forward looking. My initial ideas were to build up a profile based on opposites. Where would the writer lie on a Forwards-looking/ Backwards-looking scale, or an Assertive/ Submissive scale.

I still have some of my notes from this part of the program. Lots of scribbled pages to remind me what different aspects of the handwriting are supposed to denote. For example:

PRESSURE:

Light: Agile mind, Sensitive, Adaptable

Medium: Balanced, Adaptable

Heavy: Energetic, Outgoing, Reliable, Conflicts, Inhibited

Mixed: Emotional, Changeable, Moody

Scoring was a relatively simple process. For light pressure, for example, these would be:

```
Agile = Agile +1
Sensitive = Sensitive +1
Adaptable = Adaptable +1
```

These variables are meaningless on their own. It was also necessary to know how often a variable could potentially have been found in a handwriting sample. Thus there were also lines:

```
Agile_max = Agile_max +1
Sensitive_max = Sensitive_max +1
Adaptable_max = Adaptable_max +1
```

For the final analysis these attributes were converted into a scale out of 10, but with a proviso that to form part of the analysis an attribute would have to be found in at least 3 different tests.

Early working versions of the program contained masses and masses of tables, none of which are included in the final version. The tables were long. One started with "Abrupt", "Accurate", "Adaptable", and after some 70 attributes ended with "Unrealistic", "Unstable" and "Withdrawn". The tables enabled me to make judgements about which attributes could be safely measured and which occured too infrequently to be statistically significant. The final list, compiled by combining some attributes and omitting others, has 37 different personality traits. By now I realised I need no longer to restrict my results to scales of opposites, but that I could also indicate which of 37 personality traits were present in the writer, and whether or not these were strong.

As I tried to make sense of the tables and translate them into words, I realised something a little bit frightening was happening. I was slowly distancing myself from the books and was working on my own interpretations. The attributes were just words that had to be given a meaning and I was clarifying that meaning. Just what is meant by the word "calm"? All the people whose handwriting I had analysed and who scored high on "calm" were in my opinion for from having "calm" personalities. What they had in common, however, was that they did not readily show their emotions. Or again, what is assertiveness or submissiveness? The office manager who terrorises his staff may be as mild as a lamb in the presence of the Managing Director.

Even after I had decided on the

final form of the analysis, the program had to be calibrated to determine the cut off points between a strong, a weak or a non-existent trait. I had to find a balance between displaying too few and too many personality attributes, which I did by trial and error after analysing numerous handwriting samples. Two attributes, intelligence and adaptability, scored highly in practically ever sample I analysed and I had to reduce the sensitivity of these. I still have my doubts about the sensitivity of the latter especially as it can give some inconsistent results in some samples.

The results of an AUTO-GRAPH analysis are displayed over two screens. The first gives the scales of opposites and the second the individual characteristics that were found sorted by their strength. Statistically there are 65,536 different possibilities on the first screen and over 137,439,000,000 on the second. In theory at least each handwriting sample will have a unique analysis. Whether this is so in practice and whether I have succeeded in the task I had set myself is for the AUTO-GRAPH user to judge.

A short Visit of XMenu - Part 4

Jérôme Grimbert

Not just a question of style

Due to its history, in C, there is something called style. There is the K&R style, which was the original first C, and there is ANSI style, which is the standard C. The difference between these two style are rather tiny and unimportant, because the standard C made a provision for backward compatibility with the K&R style.

Nevertheless, there is one big difference which might hit your PE programs: the header of definition of a function.

In K&R style, you only define the returned type and the function name, with just some names for the parameters, but the types of the parameters are described after

```
int main(argc,argv)
int argc;
char** argv;
{
```

In ANSI style, the type of the parameters are in the list of the parameters.

```
int main(int argc, char** argv)
{
```

So, you would think that's not really such a big difference.

But in fact, it has a big impact when interfacing with assembly (which is how we are using the PE), because parameters which are not of type 'int' or 'pointer' are stacked differently for the function call. With K&R style, a char or a short will get allocated a full int (thus wasting some bytes on the stack), whereas in ANSI style the stack will be more compact.

If you tried to make the demo of PE application with C68, you might have wondered why some basic applications work fine, whereas the more advanced one (such as the one where there is an application window and clicking on it leave a small dot where you click) did not worked correctly.

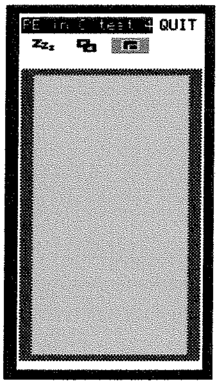
The reason is that the examples where made in K&R style, in an ancient time when C68 did not support ANSI style. With the library that was then with C68, it worked. But with the latest C68 and new library, it failed unless you convert to ANSI style the PE functions used in wrapper.

It's not very difficult, but you just have to know that unless you try endlessly to debug the sources.

In fact, not only the definition must be ANSI style, but also the declaration, especially if you have them in some header files.

So, just be aware of this evolution when making your program, and now back to the plain code.

Adding an application sub-window



I do not know if you like full listings, but usually more than two pages of them bore me and I do not even try to read it. So, because the source files are starting to getting longer each time we add something, and the interest is limited to the new thing, I will now only provide the difference between two consecutive steps.

Adding an application window is really easy.

```
--- cpe3_c
+++ cpe4_c
@@ -11,13 +11,13 @@


 char _conname[] = "con_2x1a0x0";
 /* mask startup problems, for old one */
 char *_endmsg  = NULL;
 /* and stop when I say */

-char _PROG_NAME[] = "PE in C tutorial 3";
+char _PROG_NAME[] = "PE in C tutorial 4";
 static QD_TEXTI(quit,"QUIT");
-static QD_TEXTI(title,"PE in C test 3");
+static QD_TEXTI(title,"PE in C test 4");

 static long ACTION_QUIT(struct WM_wwork
  *wwk,struct WM_litm *li);
 struct WM_action action_quit = { JSR,
  wm_actli, ACTION_QUIT};
 static long ACTION_MOVE(struct WM_wwork
  *wwk,struct WM_litm *li);
 struct WM_action action_move = { JSR,
  wm_actli, ACTION_MOVE};
```

Just changing the application name and the title of the window.

```
@@ -83,10 +83,12 @@
 {
    struct WM_wwork * result;
    struct WM_litm *loose_list;
    struct WM_infw *infw_list;
    struct WM_info *info_list;
+   struct WM_appw *aw;
+   struct WM_appl *al;

    info_list=(struct WM_info *)
     malloc(sizeof(struct WM_info)*2);
    info_list[0].xsize=14*6;
    info_list[0].ysize=10;
    info_list[0].xorg=0;
```

We will need two more pointers when setting up the window definition. aw will be the application window itself, while al will be a list of pointers (containing in our case only one pointer: aw !).

```
@@ -160,10 +162,32 @@
    loose_list[3].pact=&action_sleep;
    loose_list[3].item=3;

    loose_list[4].xsize=-1; /* end of list */

+   aw = (struct WM_appw *)
     malloc(sizeof(struct WM_appw));
+   aw->xsize=20*5;
+   aw->ysize=180;
+   aw->xorg=10;
+   aw->yorg=40;
+   aw->flag=1;
+   aw->borw=4;
+   aw->borc=255;
+   aw->papr=4;
+   aw->pspr=NULL;
+   aw->draw=NULL;
+   aw->hit=NULL;
+   aw->ctrl=NULL;
+   aw->nxsc=0;
+   aw->nysc=0;
+   aw->skey=K_TAB;
+   aw->pstat=NULL;
+
+   al = (struct WM_appl *)
     malloc(2*sizeof(struct WM_appw *));
+   al[0].pappw= aw;
+   al[1].pappw=NULL;
+
    result = (struct WM_wwork *)
     malloc(sizeof(struct WM_wwork));

SetWindowColour(result,DefaultColourSet());
    result->wstat=NULL; /* filled later */
    result->chid =0;
    result->pprec=NULL;
```

Getting and filling the structure of aw, and making the al list (which must be NULL terminated).

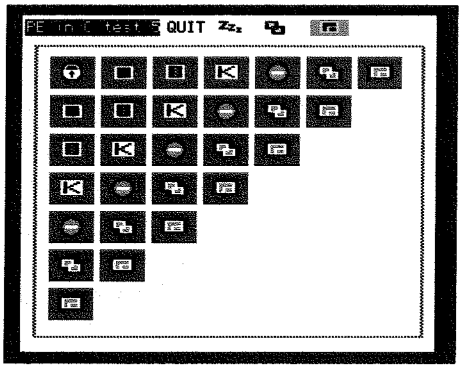
```
@@ -173,11 +197,11 @@
    result->spar3=0;
    result->pulld=0;

    result->splst=NULL;
    result->xsize=20*6;
-   result->ysize=30;
+   result->ysize=30+200;
    result->xorg=20; /* initial position of
     mouse */
    result->yorg=8;

    result->flag=1;
    result->borw=1;
```

Just updating the size of the main (primary) window, to have enough space for the application sub-window.

```
@@ -197,12 +221,12 @@
    result->pinfo=infw_list;

    result->nlitm=1;
    result->plitm=loose_list;

-   result->nappl=0;
-   result->pappl=NULL;
+   result->nappl=1;
+   result->pappl=al;
    return result;
 }
```

And of course, we need to make aware the primary window of this application sub-window. So we change the actual number of application window, and give the pointer to the list.

A menu sub-window instead



We will continue here next issue!

P-Word Review

by Dilwyn Jones

P-Word is the understated name for this potentially very useful large list of words from Paul Merdinian and RWAP Software.

Basically, it's a list of half a million English words. Yes, I said half a million words.

It can be supplied on CD-ROM, in which case you get both a QTYP spelling checker dictionary and a plain text word list if you wish to use it for other applications. Alternatively, the QTYP dictionary version can be supplied compressed on HD floppy disk, although when expanded the dictionary will only be usable on ED disks or hard disks.

I ran it through my Wordscheck program to count the actual number of words and to my cost that revealed a number of flaws in Wordscheck which I duly recompiled and now seems to cope with this massive file!

Paul Merdinian sent me the plain text version split into 6 files spread over as many floppy disks (HD disks at that). Here's a list showing the number of words in each file and the file sizes to show you just what hard work has obviously gone into this list:

FILE	WORDS	SIZE
moreac	118,797	1,249,522
moredj	127,376	1,350,243
moreko	87,825	921,380
morepr	81,480	901,912
morest	92,841	976,876
moreuz	56,099	596,221

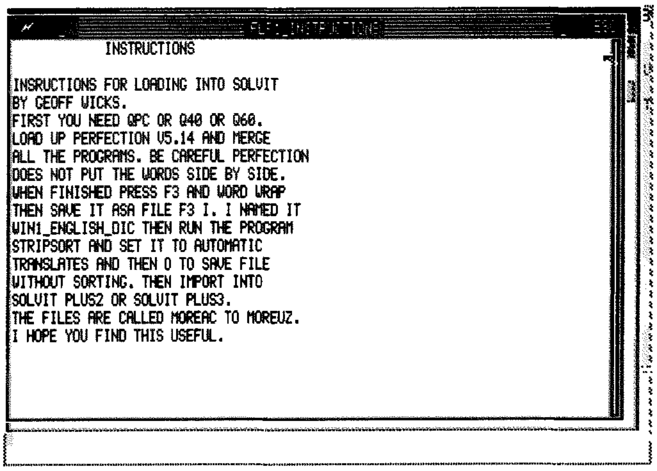
I ran a fairly simple SBASIC file merge program to join these into a single file of 564,418 words - it ended up as a file of 5,996,153 bytes. I foolishly did this on a computer in the office at work. While the process worked, it tied up that computer for several hours. And of course I then had to figure out how to transport that file from the computer at work to my home machines as it would not be a good idea to email a file of that size, and the only removable media available were floppy disks. Ho hum, eventually managed to compress the file to fit a HD disk using Zip and compressed as a QTYP file. In other words, I found out the hard way what work Paul and Rich Mellor have put into this project.

```
100 REMark merge text files listed in data statements
110 RESTORE
120 CLS : CLS #0
130 INPUT #0,'Filename to save merged files > ';op$
140 AT #0,0,0 : PRINT #0,'Words:';0
150 OPEN_NEW #3,op$
160 READ number
170 FOR a = 1 TO number
180   READ f$ : PRINT f$
190   OPEN #4,f$
200   BGET #4\FLEN(#4)-1
210   REMark files to be merged MUST end with LF
220   IF CODE(INKEY$(#4)) <> 10 THEN BPUT #4,10
230   BGET #4\0
240   REPeat loop
250     IF EOF(#4) : EXIT loop
260     INPUT #4,t$
270     IF t$ <> '' THEN
280       no = no+1
290       REMark show linecount every 100 lines
300       IF no MOD 100 = 0 THEN AT #0,0,6 : PRINT #0,no
310       PRINT #3,t$
320     END IF
330   END REPeat loop
340   CLOSE #4
350 END FOR a
360 CLOSE #3
370 AT #0,0,6 : PRINT #0,no
380 REMark number of files to merge?
390 DATA 6
400 REMark filenames of the files to be merged
410 DATA
```

As I had been supplied with the plain text file version, I decided I'd go through the process of importing the word list into QTYP to see what it involved. I found I had to use the Qtyp Dictionary Editor program (qtyp_ded) and select a template dictionary - this gives the basic format information such as language dependencies for best compression and so on.

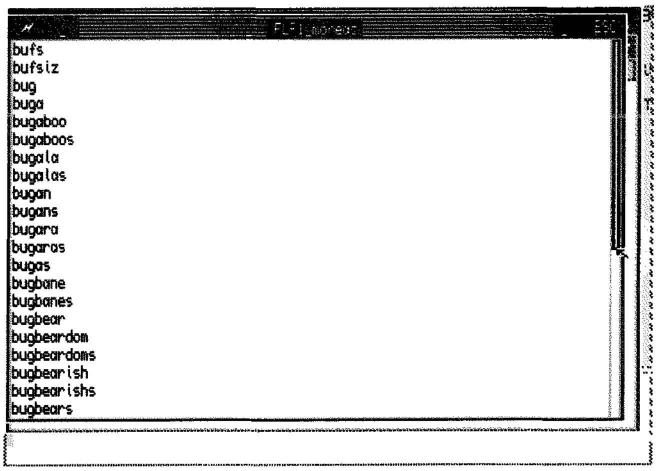
Once you have started the QTYP_DED program, create a dictionary based on the existing one (in this case an English language dictionary). This will be empty when created. Then, F3 for CMD (commands) menu, click on Load Secondary Word List, a plain text file can be loaded here. This takes quite a while to load! Once it has eventually loaded, go back to the F3 CMD menu and click on Merge Into Dictionary. Then click on Save Dictionary To File and give it a filename. Indicate if you want to compress the dictionary or not before saving. Compressing and Saving may take quite a while for a file of this size. Finally, quit from QTYP_DED and in theory at least you now have a usable half million word QTYP spell check dictionary. Using QTYP_DED on QPC2 on a fairly fast office PC running Windoze 2000 it took me about 2 hours to end up with that QTYP dictionary files. Rich Mellor says he does not now if it is possible to use this word list with the Perfection spell checker but suspects it may not be able to handle a file of this size. I considered trying Spellbound but decided that the effort versus the number of people still using Spellbound may not be worthwhile.

The P-Word plain text list can also be imported into a dictionary file for the Just Words Solvit Plus program. Solvit copes admirably with the size of the word list, although importing and creating the dictionary took quite a while. For solving word puzzles etc it proved to be a great move, as the program coped well with searching such a large wordlist and was a bit faster than I'd expected with its searches. This was one situation where P-Word came into its own. It includes names and some very unusual and uncommon words, very useful for those cryptic word puzzles you get in some papers, books and magazines. The dictionary may also come in useful for the soon-to-be-released Q-Word game from RWAP Software.



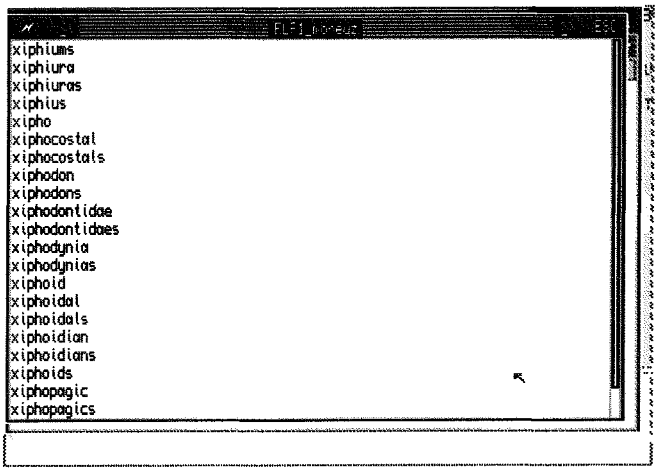
```
                INSTRUCTIONS

INSRUCTIONS FOR LOADING INTO SOLVIT
BY GEOFF VICKS.
FIRST YOU NEED QPC OR Q40 OR Q60.
LOAD UP PERFECTION V5.14 AND MERGE
ALL THE PROGRAMS. BE CAREFUL PERFECTION
DOES NOT PUT THE WORDS SIDE BY SIDE.
WHEN FINISHED PRESS F3 AND WORD WRAP
THEN SAVE IT ASA FILE F3 I. I NAMED IT
WIN1_ENGLISH_DIC THEN RUN THE PROGRAM
STRIPSORT AND SET IT TO AUTOMATIC
TRANSLATES AND THEN 0 TO SAVE FILE
WITHOUT SORTING. THEN IMPORT INTO
SOLVIT PLUS2 OR SOLVIT PLUS3.
THE FILES ARE CALLED MOREAC TO MOREUZ.
I HOPE YOU FIND THIS USEFUL.
```

Where P-Word proved to have a weakness, though, was in terms of its usefulness as a spelling checker. Most spell checkers have vocabularies of about 20,000 to 70,000 words, the more commonly used elements of a language. A list of half a million words makes it highly likely that a misspelt word will in fact be flagged as a correct spelling since there is a high chance that the incorrect spelling will be a name or other obscure word with a slightly different spelling. A typo may well actually make another correct word (a simple example might be that you intended to type the word "though" but actually typed "through" which is the wrong word, but a correct spelling).



```
bufs
bufsiz
bug
buga
bugaboo
bugaboos
bugala
bugalas
bugan
bugans
bugara
bugaras
bugas
bugbane
bugbanes
bugbear
bugbeardom
bugbeardoms
bugbearish
bugbearishs
bugbears
```

The QTYP dictionary I created with the 564,418 words amounted to the following file sizes, though this may vary a little from the version supplied by RWAP Software as it depends to some extent on the template dictionary and compression used.



```
xiphiums
xiphiura
xiphiuras
xiphius
xipho
xiphocostal
xiphocostals
xiphodon
xiphodons
xiphodontidae
xiphodontidaes
xiphodynia
xiphodynias
xiphoid
xiphoidal
xiphoidals
xiphoidian
xiphoidians
xiphoids
xiphopagic
xiphopagics
```

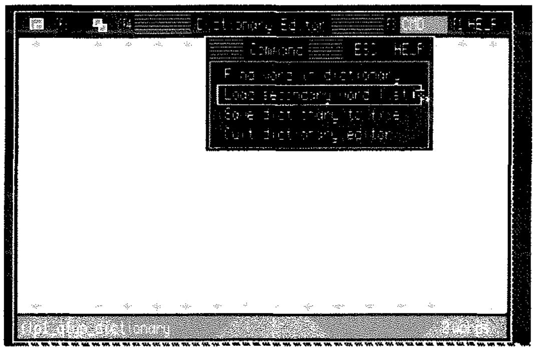
qtyp_dictionary_moreaz = 1,771,582 bytes
qtyp_dictionary_moreaz_compr = 1,086,616 bytes

In order to get these from the office computer to the home computer, I zipped them onto floppy disk, even zipped they came to 689,274 and 674,623 bytes respectively.

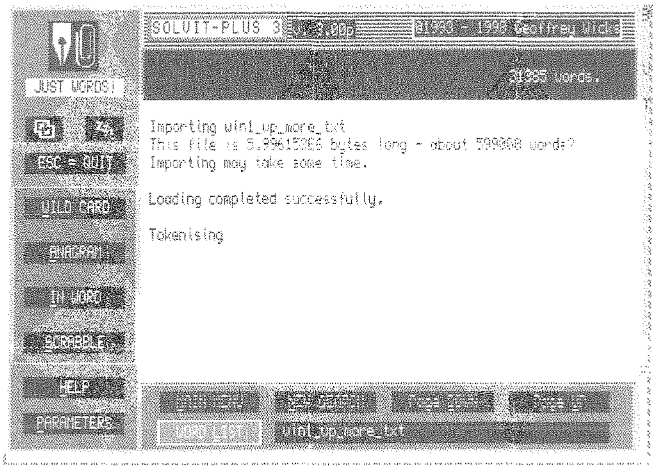
The clear implication here is that P-Word is not for older QL systems with only, say, 128K or 640K memory. RWAP Software say that a 4MB Super Gold Card is a minimum requirement. I'd agree with that, and not just because of the memory requirement. The QL's speed is a factor too when handling a dictionary of this size.

Another limitation came to mind when I started manipulating this word list. Very few of the QL text file viewers, including my own Viewer and MiniView packages could cope with a massive text file of this nature. P-Word has a linefeed between each word in the plain text version, so as far as an editor or viewer is concerned, it's a text file with half a million or more lines. Some file handling programs could cope with P-Word when viewed via their File View menus - QPAC2 Files menu and Q-Trans view menus could cope, though editors and viewers like

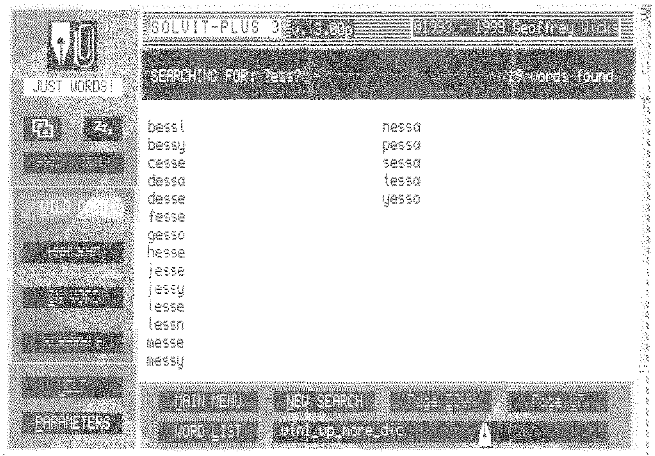
QD, S-Edit, MiniView, QED and a few others all fell over either while loading the file or shortly after loading (e.g. QD seemed to manage to load the file but as soon as you started scrolling through it froze). Perfection should be able to cope judging by the instructions file supplied - see below. With this in mind, P-Word inspired me to start writing a new pointer driven file viewer called PtrView, based on the file viewer in Q-Trans, which should be capable of scrolling through extremely large text files when complete. More news in the news pages when the program is nearing completion.



The disks came with a short text file explaining how to use Perfection and Stripsort to merge the text files into one long file and save as a single file. As I am not a Perfection user, I have to take this at face value!



```
Importing win1_up_more_txt
This file is 5.99615266 bytes long - about 599000 words?
Importing may take some time.

Loading completed successfully.

Tokenising
```



```
SEARCHING FOR: ?ess?                     19 words found

bessi                    nessa
bessy                    pessa
cesse                    sessa
dessa                    tessa
desse                    yesso
fesse
gesso
hesse
jesse
jessy
lesse
lessn
messe
messy
```

The main value of this package is the content itself, which seemed to be very good. It is obviously a very specialist package which will have limited appeal, but where it is needed it should prove extremely useful. Certainly, I have made plenty of use of it via Solvit Plus for word puzzles and so on, but have not managed to find it all that useful for spell-checks as it is simply too large to be practical as a general purpose spelling check system. I think I have learned quite a few new English words while browsing through this package!

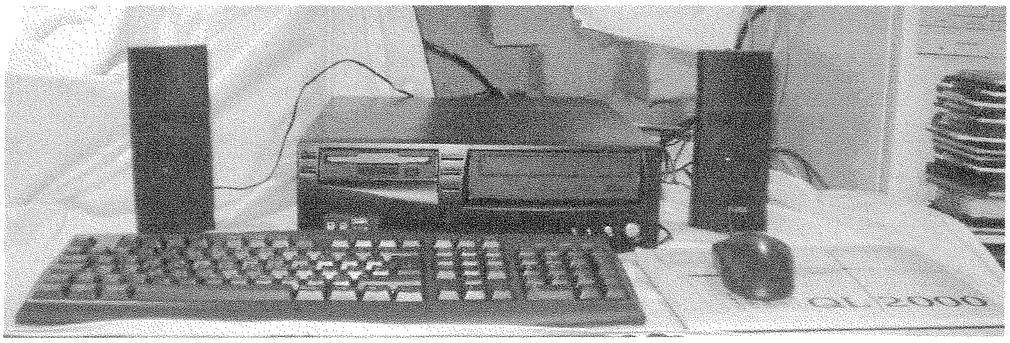
P-Word is available from RWAP Software as a QTYP dictionary file on HD floppy disk for £10 or on CD-R including the plain text file wordlist for £15.

Since this review was written, RWAP Software have contacted us to say that the actual release version will have renamed files and will be slightly smaller due to the fact that a few small corrections have been made in the release version compared to the version used for the review. For example, the compressed QTyp dictionary is now slightly shorter at 1,067,039 bytes. The CD-ROM version will now include the Solvit Plus version of the dictionary and possibly a free copy of Solvit Plus (it is now freeware). RWAP Software will look at the possibilities of converting the dictionary to other spell-checker formats if there is demand and information on the dictionary format is available.

The Outlook is Blaq - also available in white!

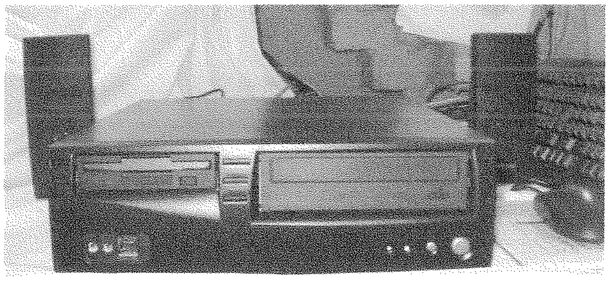
Roy Wood

My thoughts about black PC cases in the last Byts of Wood got me interested in building something to take to shows and, when we were offered black keyboard, mouse and speaker sets at a reasonable price I found the temptation hard to resist. The trouble was - what should I put into it? The case itself is very small and the power supply is, therefore non standard. ATX power supplies need adaptation before they can be used with an Aurora or Qxx system and, since the backplate of most modern boxes is set up for the ATX format I/O ports, I would need to do a bit of metal work if I was to be able to use either of these systems. None of these things are insurmountable but I had not allowed myself a lot of time to do the construction. I wanted it to be ready to show at the London QL workshop. This was on Sunday and it was already Friday. I decided, therefore, to build a PC and use QPC2.



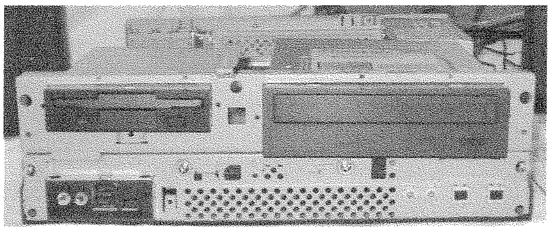
You may think that I should not be writing about this in a QL magazine but the ultimate aim was to use this for QL purposes. Some of you may have similar requirements so here is the 'how'.

Friday night arrived and I had all of the bits laid out on the kitchen table. More problems. My wife was out for the evening - she teaches drama and she took the whole of the group to see a play. Not too bad though, I thought. Put the kids to bed and I can have it done. My two year old had other ideas so I did not set screwdriver to screw until well after eleven. The whole process was so simple though that I had it up and running by 12.30 and I was installing the O/S.



It Might Seem Out of Place But.....

No one can deny that computers have come a long way from the early days of Sinclair and only a fool would try to convince you that the QL, in any of its current forms, can stand up to the abilities of the mainstream computers of today. That said it can still be a lot more fun to play with and, in many cases, it is a lot more stable. This may be, of course, because the processes that go on in the QL's O/S are a lot less complex than that of the modern PC and there are fewer people trying to crowbar extra drivers and system files into it while it is running. This is, therefore, essentially a QL project. I am going to tell you how to build an all black PC in a tiny case which can run Windows, QPC2 and LINUX. The best part of it is it does not break the bank. Everything that most people want to do with a computer can be done by this machine on one of the O/Ss mentioned above and it will leave you with enough money in your pocket to buy a flat screen, printer, scanner or any other peripheral that you may fancy.

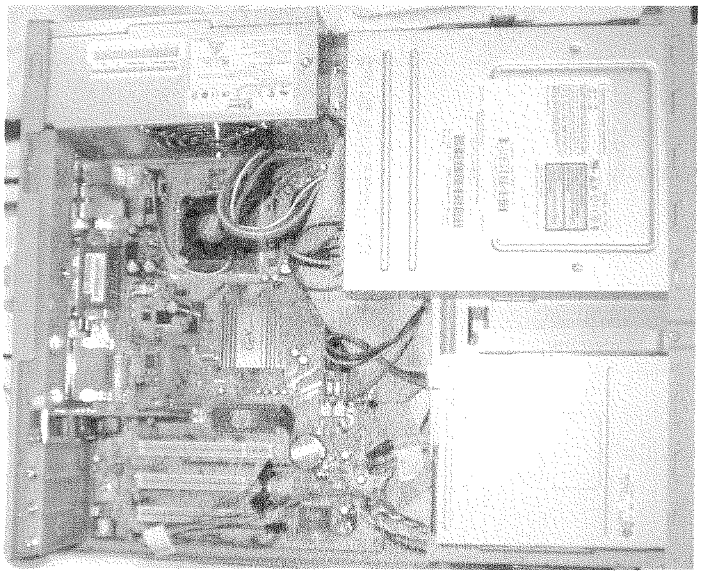


It may seem a bit sacriligious to some of you but to me it seems to be a very logical elegant space saving solution. If I compare this with the behemoth of my full tower cased P4 system it looks very small but it performs very well.

So Lets begin.

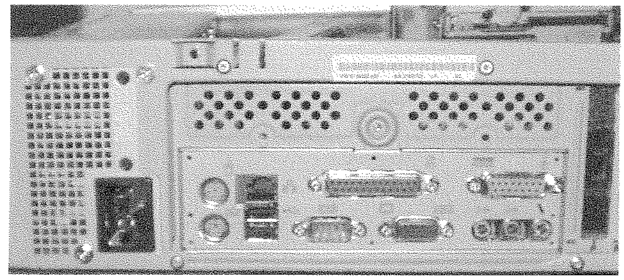
It is Not So Hard

Building a modern PC is nowhere near as difficult as it used to be. In order to put this one together I used the Eagle Tech M787CL+ Rev. 3.0A Motherboard. This unit has most of the things you might need either on board or in the box. It has a VIA C3 Samuel 2 1Giga Pro 667MHz CPU with a 133MHz FSB, 128k L1 and 64k L2 Cache. The CPU is MMX and 3DNow! compliant and has a 5200rpm Ball Bearing Fan. The board uses the SiS 630E Chipset and has 3 PCI and 1 AMR slots. It will accept 2 PC133 DIMMs adding up to 512 Mb of memory. It also has 128-bit 2D/3D VGA graphics and built in USB and Ethernet connections. All of this is right there on the board itself and the only thing you need to slot onto it is the supplied 56K modem. Having slotted the RAM onto the board - I chose to use 256 Mb - you are ready to screw it into the case. The main concern here is to make sure that the posts on the case line up with the board. Try to get one post for every hole on the board itself and count the holes and the posts. If there are more posts then holes or if some of the holes do not line up with the posts then remove any redundant ones. This is because they could short the board to earth and cause major damage. It may seem a rudimentary or simple thing to say but people have come back to me at work having put in one too many posts and destroyed the board, processor, or ram - or all of it!



With the board in place most of the work is done. The tray which will hold the the CD ROM and floppy drives hinges forward and the hard drive slots into a special bay underneath. The floppy drive must be screwed in first because the CD ROM will obscure some of the fixing holes. Once that is done You can fit the CD ROM by using screws underneath the drive tray. Connect the cables between the drives and the motherboard and power supply and plug the large ATX power connector onto the board and you are very close to finishing the task.

One major problem with these 'Flex ATX' style cases is the way that the PCI slots appear on the back. The case is only half the height of a normal desktop case which means that anything you put into the PCI slot has to be a half height or low profile card. Many modern cards are like this now, given the amount of features that they cram into the chips, but the card also has to have a half height backplate as well. The modem provided with the board is the right height for the case but plate is not so I had to take a hacksaw to the metal part and then bend it to fit. This way not really that difficult although you do have to be careful not to damage the circuit board. The board has been pop rivetted and not screwed onto the back plate so there was no safe way to remove it. The operation proved quite successful and the modem worked fine after surgery.



The last thing to do before power up is to add the front panel connections. This is where the very sparse 'manual' falls down with a vengeance. Although they number the pins and tell you what they are there are no diagrams and no numbers on the board. I had to use a bit of guesswork to get the 'power on', reset, Power LED and HDD LED cables connected. However, with all that in place, it is time to switch on.

Optional Extras

The case comes with a small pop out space at the front. Removing this (it was so 'QL' popping out a little black panel!) reveals space for a front mounted bracket with two USB ports and audio in and out. AOpen sell a little 'daughter board' to go into this slot and, to my complete amazement, the two cables that come from it just plug onto the motherboard. This gives you the added convenience of front USB connectivity.

Ignition

Turn on the switch and .... nothing! OK go back and check. All connected OK, what can this be? After a bit of rummaging around I discovered that the 'clear BIOS' jumper was set. A few years ago most boards were supplied like this but the trend died out. Let me explain. There is usually a jumper somewhere near the BIOS battery. If you set this to 'ON' it will clear the BIOS settings and make the board inoperable. This is a, kind of, last resort if you mess the BIOS up so badly the board will not fire up.

OK now it is set correctly the board fires up and sees all of the drives. Installing the operating system is simple these days. I chose to use Windoze XP for this machine because it is easy to use and (relatively) cheap if you don't buy it in a box (i.e. buy the OEM version). All you need to do is to put the CD ROM in the drive and reboot. The CD will ask a few questions, ask for the Key Code and the install with no further ado. Twenty minutes later and the system is up and running.

One of the big problems with Windoze (and I did say 'one') is that it is never quite right. At any given time there are always updates to do. In a way you can forgive them because it is a major task to maintain such a mountain of code. I am lucky here because I have ADSL so as soon as it is up and running I log onto the website and do the updates. Now for a real O/S!

QPC2 is go!

QPC2 installation is a real doddle! You can just stick the disk in the drive and run the 'install.exe' file. A short while later you have QPC2 running. In my case, however, I decided to connect the machine to the network and do it that way. I logged onto my main machine and copied all of the files across, including the QXL.WIN file on drive C:. Having done this, a matter of a few minutes on a 10/100 network, I can create a shortcut to the QPC2.exe file and fire up QPC2. I then have an identical setup on my new machine.

On a 667 MHz machine QPC2 runs pretty sharpish. I am not going to tied up in comparisons and benchmarks here because I really do not think that speed is what it is all about on a QDOS/SMSQ system. There may be a few of you out there who have applications which need fast access and run times but, even then, what this system lacks in MHz muscle it more than makes up for with its ATA 100 drive access which far faster than any system we have available. For many of the people I have spoken to the key factors are space (not having too many systems and monitors) reliability and flexibility and this system has all of that in spades.

Final Comments

This is not a high performance, all singing, all dancing PC by any stretch of the imagination but it does do the main job in hand which is to run QPC2. It also comes with a copy of the free Office suite which will allow you to view and edit Microsoft files create by their overblown and overpriced Office suite. Having a 56k Modem and and 10/100 LAN on board made it ideal for internet. Network access and USB provides connectivity to modern scanners and cameras.

Whatever way you look at it you are not going to have USB or Firewire on a QL or Q40/60 and even the ethernet connection is, as far as I know, not available to the QL side of the machine. You can use LINUX to get at the Ethernet but even that is not going to find you an ISA USB card.

You can use LINUX on this board as the main O/S and avoid Microsoft altogether but, with a decent sized hard drive, you can install LINUX as well as Windows and have all three alongside each other.

I did not intend to write this article as an advert for this system, merely to tell you all how easy and cheap it is these days. I did, however, get a lot of interest at the London Show so I am going to be offering these machines to my customers. I really think, though, that you can, and should, build it yourself. We did start off as tinkerers didn't we?

"Out of Range" or PE Windows Tamed

George Gwilt

A wave of despair threatens to drown me when once again I read the message "out of range" as the PE program I am testing kills itself. The trouble appears to be something to do with the "window definition". But what? The message "out of range" is singularly unhelpful. If the message also contained a note of which part of the labyrinthine window definition was causing the problem my despair would be considerably eased. As it is I alter the width of the thirteenth sub-window by 2 pixels and try again. A week later, after many such attempts the message is still appearing. "Out of range."

Well, perhaps that was a slight exaggeration, but nevertheless it gives a flavour of the problems that arise if a PE window definition is in some way faulty. Now, since I use a new method of defining windows, these problems have disappeared.

Window Definition

The data needed for a full description of a PE window is organised in sections linked together. At the top there is the section for the main window itself with sizes and colours and also pointers to three other sections:

* loose items
* information sub-windows, and
* application sub-windows.

The sub-windows themselves can have pointers to lists of their objects.
In whatever language a window definition is produced, it always boils down to a long list of numbers, such as the size of something, and pointers connecting the sections together. It will be no surprise then that in assembler code you might have:

```
        SIZE    14*6,10
```

where SIZE is a macro.

In C, the same information might appear as (see Jerome Grimbert's article in QL Today Vol 6 Issue 5 page 44):

```
        info_list[0].xsize = 14*6
        info_list[0].ysize = 10
```

The corresponding lines for the S*BASIC extensions QPTR might be:

```
        size%(0,0) = 14*6
        size%(0,1) = 10
```

If all the information for the window definition is valid, there is a chance that the PE program will run. Otherwise trouble will occur. When designing a window for use in non-PE programs I have often used squared paper to set out the text included in it. By numbering the rows and columns on the sheet of paper it is easy to code the ATs needed to print the information in the right place, and to see what size of window is needed for all the information.
Objects in PE windows have to be placed with pixel precision. In other words you have to think of using CURSOR and not AT for positioning. This added complexity increases the chance that a window definition will not be correct first go. There are other pitfalls too. PE does not take kindly to odd x-values. Also you have to remember the distinction between the size of a sub-window including its borders and excluding its borders and correctly set the sizes in the window definition.

SETF

To make life easier I now use the program SETF from TurboPTR to set my PE windows. This program allows you to see all the loose items, sub-windows and their objects being sized and placed. Once the operation is complete, the window itself is displayed as it will be in a PE program and the total window definition, complete with the text and sprites needed, is coded as numeric DATA lines for an S*BASIC program in a file called ram1_‹name›_WDA.

A more detailed account of SETF's procedures is given in the Appendix.

While SETF operates it does not allow loose items to overlap, does not allow sub-windows to be too small to hold the largest of its objects and it even forces enough space for a scroll bar in application windows. It automatically arranges for arrows and a scroll bar if the number of items in an application's menu is larger than can be accommodated in the application window size chosen by the user. It also automatically takes note of borders and sets all the appropriate sizes and origins.
The _WDA file produced by SETF is the coded form of three things:

* The text items needed
* The sprites needed
* Numerical information about the window

It is the last of these which is the equivalent of the information set manually for the assembler, C and QPTR versions mentioned above. Examination of this would show the details of the window and can allow alteration of the window definition. Indeed there is an S*BASIC function provided with TurboPTR, F_Loader, which will expand a _WDA file into a set of files some of which contain the numerical information referred to above. These files may be altered and re-combined to form a new _WDA file. However, the numerical information is just a list of numbers with no supporting explanation, making it awkward to interpret. Future releases of TurboPTR might contain a more friendly version of this facility if there were sufficient demand.

WintoC

There is another use to be made of the _WDA file. The program WintoC, which stands for "Window to C", will convert a _WDA file into the equivalent C instructions. For example a portion of output produced by WintoC is:

```
        48,         /* xsize      */
        10,         /* ysize      */
```

As well as providing an easy means to define windows for C programs (the primary intention) WintoC effectively decodes a _WDA file for easy examination.

Turbo

Both SETF and WintoC are written in S*BASIC and compiled by Turbo. This means that they require Turbo TK code extras. In addition SETF requires TurboPTR extras. However, the new version of Turbo allows such extras to be included in a compiled program. Furthermore, the next issue of Turbo TK Code should include a run time version designed for inclusion. Thus, if there is a demand for it, versions of SETF and WintoC with all necessary extras included could be released.

Caveat

The window definition produced by SETF as described above is a standard window definition as it appears in the official QPTR manual. This "window definition" is intended to be transferred to a "working definition" by the use of PE's WM.SETUP routine. This is certainly unlike both Tony Tebby's method described in the C68 documentation and in Jerome Grimbert's method already alluded to here. This means that the output from WintoC could not immediately be used in these systems, though it should nevertheless prove useful.
I hope to explain more about this in a future article.

Footnote

The programs SETF and WintoC are available from the SQLUG website at

www.jms1.supanet.com

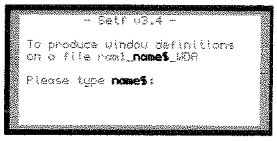
SETF belongs to TurboPTR and WintoC to CPTR.

Appendix

SETF's Operations

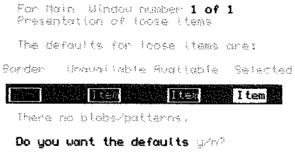
The operations of SETF are divided into five sections.
1. Name
2. Text and sprites
3. Numbers of windows and objects
4. Attributes of windows
5. Sizing and placing

1. SETF starts by asking for the name to be used for the _WDA file.



```
- Setf v3.4 -

To produce window definitions
on a file ram1_name$_WDA

Please type name$:
```

2. SETF asks for all the items of text and all the sprites, blobs and patterns which will be used.



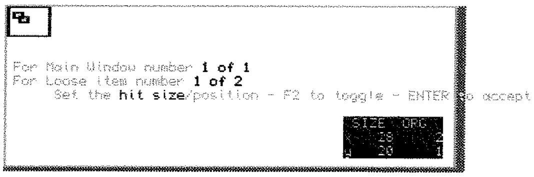
```
For Main  Window number 1 of 1
Presentation of loose items

The defaults for loose items are:

Border   Unavailable Available  Selected
                Item         Item      Item
There no blobs/patterns.

Do you want the defaults y/n?
```

3. SETF asks for the numbers of all sub-windows and the numbers of objects each will contain.

4. SETF now asks for all the attributes except size and position for all the windows. This includes such things as border size and colour, sprite to use as pointer and actual objects to be used.

When doing this SETF displays the possible items, text, sprites, blobs or patterns so that an easy choice can be made. Also, when colours have to be set the available ones are displayed.

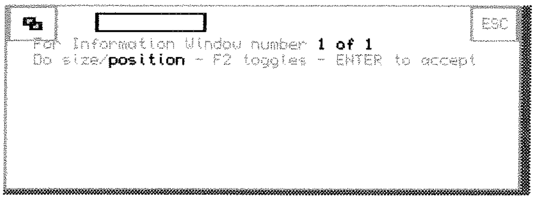
5. When all the above information has been given SETF sets about finding the size and position of all windows and objects.
First, an outline of the main window is shown. This is of the largest size possible on the screen, given the previous choice of shadow and border. The outline can be set to the desired size by using the arrow keys. The actual numerical size is given in a small window at the bottom right of the screen, so that a window size can be set either by eye or, with more precision, numerically.
Second, the initial position of the pointer within the window when it is first set up must be chosen. Again, this is done by moving a small dot to the required place. Once again the position is given in the small window at the bottom right.



```
For Main Window number 1 of 1
For Loose item number 1 of 2
     Set the hit size/position - F2 to toggle - ENTER to accept

                                          SIZE  ORG
                                           28     2
                                           20     1
```

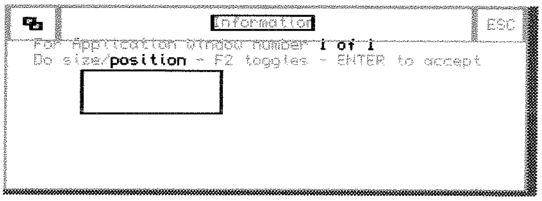
Third, the loose items are presented in order, appearing first at the top right of the window. Each has an outline, the "hit size". This size can be altered as can the position of the item. When any item is presented, the previous items' positions are also shown.
Fourth, the information windows are shown. For each in turn an outline is given, which can be resized and positioned just as for the loose items.

When a window's size and position is accepted, its objects are presented in turn for placement. When one window is complete, the next is presented.



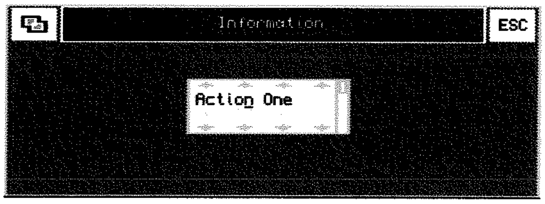
For Information Window number 1 of 1
Do size/position - F2 toggles - ENTER to accept

Fifth, the application windows are presented in the same way as for the information windows, except that this time none of their objects are presented for placing. This is because menu items are arranged in a grid of size pre-determined by the number and maximum size of the objects.



For Application Window number 1 of 1
Do size/position - F2 toggles - ENTER to accept

And this is the final window as designed by SETF:



George can be contacted via e_mail:
geo.gwilt@argonet.co.uk

QLTdis - part 8

Norman Dunbar

Well, here's a surprise, since part 7 was published in the last issue of QL Toady, I've had no emails to say that errors were found - this could be the first time that has happened. let's hope I can keep it up.

Unfortunately, this does not mean I got off scott free - I've had to write a small bug-fix routine to correct some problems I've been having with a small number of instructions, the following gives details of changes that should be made.

Code corrections

This simple piece of code ...

```
*--------------------------------------------------------------------------
* Type 19 - ADDQ and SUBQ.
*--------------------------------------------------------------------------
dtype_19    bsr     size_decode     ; Decode the size and add a space
```

becomes the slightly more complicated code that follows:

```
*--------------------------------------------------------------------------
* Type 19 - ADDQ and SUBQ. There is a problem here, if we decode an Scc
*           instruction we come here instead of type 26. Test bits 6&7 for '11'
*           and if found, skip to type 26 rather than here !
*--------------------------------------------------------------------------
dtype_19    andi.b  #$c0,d0         ; Keep only bits 6 & 7 or the low byte
            cmpi.b  #$c0,d0         ; If both bits set, this should be a type 26 ?
            bne.s   t19_not_t26     ; We're ok - skip over type 26 stuff

t19_type26  lea     t_s,a3          ; This is the correct instruction text
            bsr     fix_up_bug      ; Sort out the buffer
            move.w  d7,d0           ; Reset the op-code
            bra     dtype_26        ; And do a type 26 Scc instruction

t19_not_t26 move.w  d7,d0           ; Reset the op-code again and do type 19
            bsr     size_decode     ; Decode the size and add a space
```

And another one with the same problem. When testing type 26, in particular the CMPA.L Ax,Ay instruction, the output was actually CMPM (Ax)+,(Ay)+ which is not quite right. Again, this is caused by bits 6 & 7 being set to '11' for CMPA.L so once again, some changes are required, this time to the type 13 code at dtype_13.

Again, a very simple piece of code needs changing:

```
*-------------------------------------------------------------------------
* TYPE 13 - The CMPM instruction.
*-------------------------------------------------------------------------
dtype_13    bsr     size_decode     ; Add the size specifier and a space
```

to the following:

```
*-------------------------------------------------------------------------
* TYPE 13 - The CMPM instruction. Here we have problems as this routine catches
*           CMPA.L Ax,Ay and decodes it as CMPM (Ax)+,(Ay)+. We can test to
*           fix this by checking bits 6 & 7 and if both are set, we are in the
*           wrong place !
*-------------------------------------------------------------------------
dtype_13    andi.b  #$c0,d0         ; Keep only bits 6 & 7 of the low byte
            cmpi.b  #$c0,d0         ; If both bits set, this should be a type 25
            bne.s   t13_not_t25     ; We're ok - skip over type 25 stuff

t13_type25  lea     t_cmpm,a3       ; This is the correct instruction text
            bsr     fix_up_bug      ; Sort out the buffer
            move.w  d7,d0           ; Reset the op-code
            bra     dtype_25        ; And do a type 25 CMPA.L instruction

t13_not_t25 move.w  d7,d0           ; Reset the op-code again and do type 19
            bsr     size_decode     ; Add the size specifier and a space
```

And again, the following code has changed to that shown further below. Just when I thought the last episode was ok, I find a fault all by myself!

```
*-------------------------------------------------------------------------
* TYPE_24 the ADDs and SUBs
*-------------------------------------------------------------------------
dtype_24    bsr     size_d0         ; Get the size into D0 or 6 for ADDA/SUBA
```

The code above should be changed to the following:

```
*-------------------------------------------------------------------------
* TYPE_24 the ADDs and SUBs. We need to trap ADDX and SUBX here as well and, if
*         found, redirect to type 30 below. This can be done by checking for the
*         value 1 00 in bits 8, 5 and 4 of the op-code as this is specific to
*         the ADDX/SUBX family and cannot occur in ADD/SUB legally.
*-------------------------------------------------------------------------
dtype_24    andi.w  #$0130,d0       ; Mask out bits 8, 5 and 4
            cmpi.w  #$0100,d0       ; ADDX or SUBX ?
            bne.s   t24_not_t30     ; No, skip over type 30 stuff

t24_type_30 move.w  d7,d0           ; Retrieve the op-code again
            move.w  d0,d4           ; Working register
            andi.w  #$f000,d4       ; Keep top 4 bits only
            cmpi.w  #$9000,d4       ; SUBX ?
            bne.s   t24_addx        ; No, skip

t24_subx    lea     t_subx,a3       ; SUBX text
            bra.s   t24_both        ; Skip over ADDX stuff

t24_addx    lea     t_addx,a3       ; ADDX text

t24_both    bsr.s   fix_up_bug      ; Sort out the buffer
            bra.s   dtype_30        ; Do it properly now

t24_not_t30 move.w  d7,d0           ; Restore the op-code
            bsr     size_d0         ; Get the size into D0 or 6 for ADDA/SUBA
```

Please note also that there is a slight bug in GWASL the free assembler, since versions 1.2 onwards. I've only just found out but the MOVEP instruction is not being correctly assembled as bits 7 and 8 of the op-code are always zero. This has been reported to George and I'm sure he will get a fixed version out as soon as he can.

I did spend some time tracking this down as I thought I'd made yet another error in my code - but after about two solid hours of looking and pondering, I was no wiser - until I took a closer look at the binary code rather than the text - then it all became clear. George's assembler is getting a good thrashing in this series :o)

Having said that, I have never actually used the MOVEP instruction in all my years of coding 68000 assembly language so this bug is unlikely to affect anyone following this series. In fact, the only time I had ever heard of it being used is in the SMSQ/E source code - and it is currently being removed from the Q60 version of the source anyway.

On with the code

Now that we have the obligitory corrections out of the way, the remainder of this instalment continues with the type 26 instructions. Considering the troubles I had with this instruction, there isn't really much to it.

First of all, mask out the condition code in bits 8 to 11, then shift it down to bits 0 to 3 and call the sub-routine which puts the appropriate condition code into the buffer. After adding a space to the buffer, we reload the op-code (seeing as how we trashed D0) and extract the effective address. Simple - once you get to the correct part of the code that is!

```
*------------------------------------------------------------------------
* TYPE 26 - the Scc instructions
*------------------------------------------------------------------------
dtype_26    andi.w  $0f00,d0       ; Extract the condition code in bits 8 - 11
            lsr.w   #8,d0          ; Shift down to bits 0 - 3
            bsr     cond_code      ; Decode the condition code
            bsr     space          ; Add a space
            move.w  d7,d0          ; Reload the op-code
            bsr     eff_addr       ; Effective address decode
            bra     p_hex          ; Done
```

The next family is the type 27 or MOVEM instructions. As I'm still debugging the register list decoding routines, I'm saving that bit of code for the next instalment. So, don't worry about the missing code for now - all will be revealed next time.

```
*------------------------------------------------------------------------
* TYPE 27 - MOVEM instructions
*------------------------------------------------------------------------
```

So, having passed over type 27 for now, type 28 - MOVEA - instructions are next, there is nothing special going on here - it all just works (famous last words). All we do here is check the size and add it on to the decoded instruction, and then extract the effective address and finaly the destination register - which is always an address register.

```
*------------------------------------------------------------------------
* TYPE 28 - MOVEA instructions
*------------------------------------------------------------------------
dtype_28    btst    #12,d7         ; Check the size
            beq.s   t28_long       ; Clear is long
            bsr     uu             ; Size is 'W'
            moveq   #2,d5          ; Set word sized op
            bra.s   t28_add        ; Skip

t28_long    bsr     ell            ; Size is 'L'
            moveq   #4,d5          ; Set long sized op

t28_add     bsr     space          ; Add a space
            bsr     eff_addr       ; Extract the effective address
            bsr     comma_a        ; ',A' required next
            bsr     dest_reg       ; Extract the destination register
            bra     p_hex          ; Done
```

The next instruction family consists of the `MOVE <ea>,<ea>` instructions and these are at fist look quite complicated, but by following the code below, it can be seen that they are actually quite simple to decode.

```
*------------------------------------------------------------------------
* Type 29 - MOVE <ea>,<ea>
*------------------------------------------------------------------------
dtype_29    andi.w  #$3000,d0      ; Keep only bits 12 and 13
            lsr.w   #8,d0          ; Shift size bits into low end of D0
            lsr.w   #4,d0           ; 8 bits at a time !
            cmpi.b  #1,d0           ; 1 = .B
            bne.s   t29_long        ; Try LONG
            move.b  #'B',d4         ; Byte it is then
            bsr     str_add_b       ; Add to the buffer
            moveq   #1,d5           ; Byte sized op
            bra.s   t29_size        ; Skip

t29_long    cmpi.b  #2,d0           ; 2 = .L
            bne.s   t29_word        ; Must be word
            bsr     ell             ; size is 'L'
            moveq   #4,d5           ; Long sized op
            bra.s   t29_size        ; Skip

t29_word    bsr     uu              ; Size is 'W'
            moveq   #2,d5           ; Word sized op

t29_size    bsr     space           ; Add a space
            move.w  d7,d0           ; Reload the op-code
            bsr     eff_addr        ; Extract first effective address
            bsr     comma           ; Followed by a comma
            lsr.w   #6,d0           ; Shift the other ‹ea› into position

*-------------------------------------------------------------------------
* At this point, the source affective address has been correctly extracted,
* however, the destination is 'the wrong way round' because the mode and
* register are swapped over. We need to get them the correct way around or
* the eff_addr routine will produce garbage for the destination effective
* address.
*-------------------------------------------------------------------------

            andi.b  #$3F,d0         ; Mask out unwanted bits 6 & 7
            move.b  d0,d4           ; D0 = D4 = 00rrrmmm
            lsr.b   #3,d0           ; D0 = 00000rrr
            lsl.b   #3,d4           ; D4 = rrmmm000
            or.b    d4,d0           ; D0 = rrmmmrrr
            bsr     eff_addr        ; And extract it too
            bra     p_hex           ; All done
```

Some of the type 30 family starts decoding as the type 24 family - which is a bit of a bother. However, the alterations to the type 24 code send the errant ones down here eventually for corrective pocessing. By the time we get here the buffer has been loaded with the correct instruction (ADDX or SUBX) and the following code works out the size and whether we are adding data registers or memory addresses.

```
*-------------------------------------------------------------------------
* Type 30 - ADDX
*-------------------------------------------------------------------------
dType_30    bsr     size_decode     ; Add the size specifier to the buffer
            btst    #3,d7           ; Set = ADDX -(Ax),-(Ay)
            beq.s   t30_data        ; Must be data registers
            bsr     mba             ; Add the address stuff '-(A'
            bra     t30_reg         ; Skip

t30_data    bsr     dddd            ; Add a data register

t30_reg     bsr     src_reg         ; Add the source register
            btst    #3,d7           ; Set = ADDX -(Ax),-(Ay)
            beq.s   t30_data2       ; Must be data registers
            bsr     r_bracket       ; Close bracket next
            bsr     comma_mba       ; Address stuff ',-(A'
            bra.s   t30_reg2        ; Skip

t30_data2   bsr     comma_d         ; Do data registers

t30_reg2    bsr     dest_reg        ; Add the dest register
            btst    #3,d7           ; Again !
            beq     p_hex           ; Data registers are done
            bsr     r_bracket       ; Close bracket
            bra     p_hex           ; Done address regs too
```

The last family of instructions simply outputs a DC.W $xxxx to the buffer. It is called whenever an intruction word passes through the entire list of tests without triggering any other 'type' routines. As I have trapped all possible instruction types, anything which falls through cannot be a valid instruction, so it is most likely to be data. If so, we should catch it here.

```
*--------------------------------------------------------------------------
* Type 31 - DC.W catch all for undecoded instructions - probably data.
*--------------------------------------------------------------------------
dtype_31    move.w  d7,d4
            bsr     d4_hex_w
            bra     p_hex
```

Finally, add the following at the very end of the diss_asm file. This is the small bug-fix routine which I have had to add for all those painful occasions when an instruction family has wayward members who wander off into the wrong decoding routines.

What this routine allows us to do is clean out the output buffer of its current contents, then load the correct instruction text into it ready to carry on processing as if nothing had happened - we know better :o)

I don't particularly like this fix, however, it works. Had I written the disassembler BEFORE I did the articles I would have fixed this in another way, and you would never have known the problems I've been having!

```
*-----------------------------------------------------------------------
* This routine is entered with A3.L pointing to the correct location of an
* instruction in op_table because certain decoding masks cannot be set up
* correctly - Scc as a type 26 decoded as ADDX or SUBX which is a type 19
* for this simple reason. Yes I would do it differently next time !
* This routine simply deletes the current buffer contents and replaces it
* with the data at A3 instead.
*-----------------------------------------------------------------------
fix_up_bug  movem.l d4/a1-a2,-(a7)  ; Save all working registers
            movea.l a3,a1           ; Correct FROM address
            lea     output,a2       ; To Address
            movea.l a2,a5           ; Reset the output buffer
            addq.l  #2,a5           ; First character in output buffer
            move.w  (a1),d6         ; Length of instruction text
            adda.w  d6,a5           ; Where we should be when correct !
            bsr     str_copy        ; Copy correct text to buffer
            movem.l (a7)+,d4/a1-a2  ; restore workers
            rts                     ; All done
```

Next issue we will take a look at the final remaining instruction to be decoded - MOVEM - and I might even let you in on all the stuff I had to do the get the bugs out of my decoding routines. See you then.

GD2 Programs

Dilwyn Jones

GD2, the Graphics Device 2, the so-called "colour drivers", has been around for a little while now on various QL-compatible systems such as QXL, QPC2 and Q40/Q60, so I thought the time might be right to publish a list of programs available which make use of GD2 in some way to provide us with better on-screen graphics. This list isn't meant to show which programs are compatible with GD2 (most well written QL programs will work under GD2 in much the same way as they would on older QL systems) but rather programs such as graphics programs which make direct use of the higher numbers of colours available.

Most of these programs are freeware, you should be able to get copies from PD libraries or from the various QL-related web sites.

PQIV - Graphics viewer from Claus Graf

Photon - JPEG viewer from Dave Westbury

GD2 Docs - Tony Tebby's documentation files for the GD2 system.

Prowess - the window manager system from PROGS. Applications which rely on Prowess should be mentioned here too - such as the

Paragraph word processor from F Lanciault.

BMP2PIC - file conversion program from Phoebus Dokos, converts windows BMP files into QL PIC files.

PhotoQL - Roberto Porro's graphics conversion program.

Pnm2picr (Q40) - available from the Q40 web site, I don't know much about this program.

PIC2BMP - conversion program from Jerome Grimbert.

Q-Colour. Colour picker and display system from Wolfgang Uhlig, includes the colour "skins" extensions from Wolfgang Lenerz.

Sprite Editor - from Jerome Grimbert.

QL3D1 - from Mark Swift?

PSA conversion from George Gwilt, converts partial save area files between Q40/Q60 mode 33 style graphics to mode 32 style graphics.

PCBCad from Malcolm Lear. PCB/design program.

Screen Snatcher - grab copies of the screen picture, works on both traditional QL mode 4/8 graphics plus the new modes.

PicView - image file viewer for QL screens and PIC files.

QCDEZE from Duncan Neithercutt is a CD-ROM handler front end which uses GD2 graphics on Q40/Q60 systems.

Pan and Scroll Toolkit from Wolfgang Lenerz, available on the Phoebus Dokos website.

QDT - the QL desktop system from Jim Hunkins.

Anyone know of any more? I may update this article from time to time.

Programming with QPTR - Part 3 The Level II pointers, continued

Wolfgang Lenerz

Again we continue our exploration of QPTR. You may remember that to make a window under the Extended Environment with the Sbasic QPTR toolkit, you need a working definition, and that the working definition is obtained by the function ***MK_WDEF***, thus:

```
workdef = MK_WDEF(wdef%,wattr%,wptr,ltab
,inftab,apptab)
```

Here, wtpr, ltab, inftab and apptab are 'level II pointers'. In the last instalment, we stopped at these level II pointers, and more specifically after having explained "wptr", the pointer towards a sprite definition, sprite which will be used in the window for the mouse pointer. We now know how to define sprites.

So let's have a closer look at the other Level II pointers, and first 'ltab', the loose items table, or pointer towards the loose menu itmes list.

<u>B- The "loose menu" items list</u>

or: Of menus and items.

The concept of a "menu" probably does not need much more explanation: a menu is just a set of options proposed to the user, who makes his choice either by hitting a key corresponding to the option, or by clicking on the option with the mouse. An "**item**" of a menu is simply one of the choices of that menu. In older programs (such as Quill), a menu is displayed as a regular list, such as:

F1 = action 1
F2 = action 2
F3 = action 3
F4 = action 4

and so on. This kind of menu, whilst regular, is also boring as it is generally bundled closely together in the window, and it is difficult to show, at the same time, other information in addition to the menu choices. It would be nice to have the menu items anywhere in the window, instead of having them in a regular grid as shown above.

This is what a 'loose' menu allows us to do. As the term implies, the items of such a menu are 'loose', i.e. can be anywhere in the window, they don't have to be in a rectangular grid - but they are still part of the same menu. The advantage is that, whilst the items are part of a regular structure (and thus easily recognizable), they are also placed where they can be used to best effect. The structure of the menu is regular in that the items will have the same appearance, but the items do not appear one after the other in the window. This is why it is a "loose menu".

When you define a loose menu for the window, you will have to define what each menu item in this menu is and does. Also, as the items are part of the same menu, they will have some properties in common (their general appearance). Some other properties will depend on each item (such as the key which actions each item – it would be unfortunate if that were the same for each item). You must determine all of that.

The common properties are those that define the general aspect of the items: what colour is used as the "paper" or "ink" to display the items, what type of border they will have, etc. Take for example the QPAC2 "Files" window: the menu choices offered (Command, View, All, ESC and so on) have the same general aspect (same colour, same ink, same paper, same border colour when the pointer is in them etc...): they are all items of the same loose menu. They all change similarly when actioned or when the pointer moves over them. So they all change "status" in a similar manner. Let's take a closer look, first, at item "statuses".

1 - Item Statuses

Actually, loose menu items can have four different **statuses**: The first status, is simply that of a normal item which you can hit or do. It is said that this item is "**available**". The second status is where, for any reason, you can neither "**hit**" nor "**do**" the item: the item is "**unavailable**" and cannot produce any action. The third status is that of an item over which the pointer is just hovering: this is now the **current** item, and a border is drawn around it – if you HIT or DO in the window, it will be this item that is actioned. The last status is that of a **selected** item, which is what happens when you hit an item and it stays emphasized – such as the View item in QPAC2-Files.

The definition of the four statuses will be common to all items of a loose menu. This seems logical, and avoids confusion: if red paper with black ink meant that one item was selected, but meant that another item was unavailable, this would confuse the user to no end. Thus, the definition of the colours used for these different statuses are the same for all items. This provides the regularity which enables the user to recognize loose menu items instinctively as such.

To each status for the loose items thus correspond "**item attributes**". The item attributes are common to all items, and define the paper, border and ink colours for each status.

Some other aspects, however, may be different for each item. In fact, it is as if each item had a "window" with a content. Thus, for each item, you should indicate what the size and position of its "window" should be, and also its content and type (text or sprite). You also define its "selection key" and so on. The selection key is the key you hit to have the corresponding item produce an action – F4 for the "View" item in Qpac2.

2 - Making the loose menu items list

All the data for these items is grouped together in a list called "loose menu items list" or "Loose Items List" (LIL). This list contains the common definitions for all items, and the different information for each item, one after the other. To make this list, you should use the function ***MK_LIL*** (***M**a**K**e **L**oose **I**tem **L**ist):

```
ltab = MK_LIL (lattr, lsiz%, lorg%, ljus%,
key$, ltyp%, lstr$, lspr, lblb, lpat)
```

ltab is the result of this function and is a pointer to the loose items list. The parameters are as follows:

–› * **lattr**. This is an array of dimension DIM lattr(3,3). It contains the item *attributes*. These are the different colours/borders which show the different statuses of the items. As mentioned above, these are common to all items.

- lattr (0) (i.e. lattr (0,0), lattr (0,1), lattar (0,2) and lattr (0,3)) contains, in this order, the size and colour of the border of the current item, in lattr (0,0) and lattr (0,1). lattr(0,2) and (0,3) are unused.

- lattr(1) contains the paper and ink colours for unavailable items in lattr (1,0) and lattr(1,1). The two other elements of lattr (1) (i.e. lattr (1,2) and (1,3)) point to a "blob" and a "pattern" (more of which later): in general, though, **they are left empty**. For my part, I don't think I've ever encountered a program where they weren't left empty i.e. 0 (just putting myself out on limb here, of course).

- Next, same thing for available items(lattr 2,0 et 2,1)

- Next, same thing for selected items (lattr 3,0 et 3,1)

–› * **lsiz%**, **lorg%**, and **ljus%** are integer arrays of dimension DIM (n–1,1) where n is the number of items in the menu (numbering starts at 0). Thus, if you have three loose menu items, you'll DIM the arrays DIM (2,1). Element 0 (i.e 0,0) and (0,1) then contains information

RWAP SOFTWARE

PWord English Dictionary v1.0	£15

The ultimate UK english dictionary with over HALF A MILLION WORDS, compiled by Paul Merdinian who compiled the Mega dictionary for DP's Speelchecker.
Due to the sheer size, a Super Gold Card is the minimum requirement.
Two versions are available: QTYP dictionary only on HD disk - £10 CD containing ASCII list, QTYP version, Solvit Plus (from Just Words) and the dictionary in the solvit format, cost £15.

QL Cash Trader v3.7	£5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT returns, reports and analysis for audit trails and management decisions. Previously sold for over £100.*

QL Payroll v3.5	£5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of the current tax year rules.

Sidewriter v1.08	£10
Image D v1.03	£10
Q-Help v1.06	£10
Q-Index v1.05	£5

Four excellent programs to assist the QL user.
Sidewriter: Produce landscape printouts on Epson printers.
Image D: Produce 3D pictures of objects.
Q-Help: on-screen help for SuperBASIC commands.
Q-Index: look up keywords related to topics.
See earlier adverts for more detials.

ProForma ESC/P2 Drivers v1.04	£8

New improved colour and monochrome printer drivers, providing up to 720dpi for all programs written for use with ProWesS, such as LineDesign and Paragraph. Works on all Epson inkjet printers which support binary mode compression (740, 850 and 900 models at least). 1440 dpi to follow.

QL Genealogist v3.26	£20
Genealogy For Windows	£50

Store your family tree for posterity. Add individuals with details of their parents and children, watch all of those links build up into a formal family tree layout. Text files and pictures may also be linked to individuals as well as notes and events, making this the perfect way to preserve the history of your family.
QL version now supports FileInfo II and QMenu as well as allowing you to link both male and female trees.
Sample tree of the Royal family since 1066 included.
PC version is event driven - enter the details as they appear in documents and it generates the tree from these. QL data and GEDCOM can be transferred to the PC version. Upgrade to latest PC version (v5.21) for £8
Both programs easy to use and complete with a step by step tutorial.
** QL USERS upgrade to PC version for £25 ONLY **

Q-Word *COMING SOON*	£tba

The ultimate word game for the QL - you are given a grid of letters and need to link letters together to form as many words as possible. Points are based on both the number of words and their length, as well as letters used. As you use a letter, it is removed from the grid, with the object to clear the grid.
Using high colour graphics on all systems which support more than 8 colours (including Aurora), background music and much more, this will keep you entertained for a long time.

SBASIC/SuperBASIC Reference Manual	£40
Updates £6 each, £10 for 2 (Current Version - Rel 4)	

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on each of the different QLs, plus SMSQ/e, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.
This book is ideal for all QL users and is kept up to date with regular updates. ** Currently Out of Print **

QL Cosmos v2.04	£5

Ever wondered what the stars in the sky looked like 100 years ago? Or, maybe you want to learn the constellations and names of what you see in the sky. This is the program for you - generates pictures of the stars and planets for any given place or time and provides details on these objects. Includes Halley's Comet, the Moon and the Solar System planets.

Q-Route v2.00	£25
Upgrade from v1.xx	£5

The latest version of this popular route finding program. Find the quickest route or the shortest route between any two places, using roads. A wide range of maps is available for this program (see elsewhere in this advert). The program is easy and quick to use. You can even add your own places and roads to the maps to include local detail.

Flashback SE v2.03 (Upgrade only)	£5

The ultimate database program - extremely fast and flexible, easy to use, updated to cope with the latest versions of the QL operating system and still maintained. A report module is included to allow you to format output in any way, including mail-merge. Unfortunately only available as an upgrade from the original version (original still available from Sector Software).

Return To Eden v3.08	£10
Nemesis MKII v2.03	£8
The Prawn v2.01	£8
Horrorday v3.1	£8
West v2.00	£5
The Lost Kingdom of Zkul v2.01	£5

A wealth of QL adventures - mainly text only.
Save the Galaxy from the ambitions of the evil dictator Nemesis.
Battle against werewolves and dracula look-alikes on a Hammer Horror set in the comical Horrorday.
Take the part of a prawn with a hangover, lost in a strange land in the hilarious Prawn.
Solve a bank-robbery by fighting the bad guys and collecting the loot in real-time old West.
Battle countless dwarves in the atmospheric Lost Kingdom of Zkul.
Return to Eden is a massive adventure over 3 disks with colourful graphics - control 3 characters in their quest to find the missing Prince.
All six adventures are available together for only £25.

A range of games to keep both the young and the young at heart amused. Some are old favourites, like Golf and a pub quiz program (500+ questions). Others are fast, colourful arcade games. Flight simulator also now available. Plenty of variation and skill required - what more can you ask for? All 6 programs only £28.

Open Golf v5.20	£8
QuizMaster II v2.07	£5
Stone Raider II v2.00	£5
Hoverzone v1.2	£5
Deathstrike v1.5	£5
Flightdeck v1.05	£10

The latest maps for Q-Route. Maps of various areas of Britain have been created by cutting them out of Big Britain Map - they will use less memory and can contain more detail. Areas covered: Scotland, NE England, NW England, S&W Yorkshire, Wales & Derbyshire, London area and South England.
Latest version of Q-Route is recommended.

Britain.map v1.11	£2
BIG Britain Map (needs 2MB) v2.03	£5
Various Britain Area Maps (ask for details)	£2 ea.
Ireland Map v1.00	£5
Belgium Map v1.01	£2
Catalonia Map v1.03	£2

D-Day MKII v3.04	£10
Grey Wolf v1.8	£8
War In The East MKII v1.24 (Upgrade Only)	£5

For the gaming enthusiast - D-Day is a classic table top wargame for one or two players - you control either the Allies or the Axis forces during WWII. With the ability to define your own army set ups and a choice of 4 different scenarios, this should keep you entertained for a while.
Grey Wolf is a graphical simulation of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

RWAP Software, 35 Chantry Croft,
Kinsley, Pontefract, West Yorkshire
WF9 5JH
TEL: 01977 610509
http://hometown.aol.co.uk/RWAPSoftware

Cheques in £sterling
payable to 'R.Mellor'

* Also known as Trading Accounts

about the first loose item (item 0) element 1 of the array contains information about loose item nbr. 2 and so on. Just what the information held in the elements is, is explained here:

a) **lsiz%** contains the x and y sizes of each item. One could say that these are the size of the "window" for each menu item. It is this window whose paper colour will change when the item becomes selected. Each element of this array contains, in element (n,0) the X size of the window, and in element (n,1) the Y size of the window.

b) **lorg%** contains the "origin", i.e. the x and y position of this "window" for each item. The position is given as the top left corner of the "window", in pixels, **and relative to the origin of the (primary or secondary) window containing the loose menu items**. Each element of this array contains, in element (n,0) the X position of the window, and in element (n,1) the Y position of the window.

c) **ljus%** is the x/y justification of the content of the item with respect to its "window" (i.e. if the item contains a text, is the text centered, is it flush to the left, or to the right?) The "window" for a loose item can be larger than its content, and then it is important to state where the content should be. For example, the 'F6 Sort' item in the QPAC2 Files menu generally has a window that is larger than its content, which can be seen when you move the pointer over it: the border around the item is larger than the content of the item. With the ljus% parameter you indicate the number of pixels from where the content of the item should be drawn or printed, with respect to the top left corner of the item's "window". If this parameter is 0 in any of the directions (x or y), then the item will be **centered** in that direction. Each element of this array contains, in element (n,0) the X justification of the content of the window, and in element (n,1) the Y justiciation of the content of the window.

-› * **Key$** is a string that contains the selection key for the items. The selection key for an item is the key to be pressed to hit/do the item. KEY$ is one large string made up of the selection keys for each item, so that key$(0) = the selection key for item 0, key$(1) = the selection key for item 1 and so on. Thus Key$ is composed as follows:

```
key$=chr$(n1)&chr$(n2)&...&chr$(nx)
```

i.e. exactly ONE keypress character per item, until x items. The first is for the first item, and so on. You can also write:

`key$="A"&"B"&"C"` etc...

If you do not wish an item to have the possibility to be hit/done with a keypress, use CHR$(0) in the string for the keypress for this item.

The character in question MUST be put in UPPER CASE, (i.e. either "A" instead of "a" or CHR$(65) instead of CHR$(97)). It doesn't matter, later on, whether the uses presses the key in upper of lower case, but here at the definition stage, you MUST give it in upper case.

There are also some special characters:

CHR$(1)= Hit= SPACE/left mouse button (not to be used as selection key)
CHR$(2) = DO = ENTER/Rightmouse button (not to be used as selection key)
CHR$(3) = Cancel = ESC
CHR$(4) = Help = F1
CHR$(5) = Move window = CTRL F4
CHR$(6) = Change size = CTRL F3
CHR$(7) = Wake = CTRL F2
CHR$(8) = Sleep = CTRL F1

Thus, if you have an item the action of which will move the window (it should then have the standard sprite for that, as well), the key$ for this item should be CHR$(5), and thus, each time you hit the standard CTRL-F4 combination to move the window, this item will be actioned. You COULD conceivably use any other key, but it really is better if you use the standard keypresses for these standard items!

All of these actions should be quite clear, except perhaps wake and sleep: Try CTRL F1 and CTRL F2 in QPAC 2, and you will notice that sleep puts the program to sleep as a button, CTRL F2 wakes it up again, and refreshes the menus.

Of course, you are not required to provide for buttons, wake or even window move events in your programs. If you do provide for this, however, it is suggested that you use the standard keypresses for the items concerned.

-› * **ltyp%** is an array of dimension DIM (n), i.e. one single element per item. This array determines the item type. There are four types:

0 = the item is a string
2 = " " " " sprite
4 = " " " " blob
6 = " " " " pattern

Once the type is determined to be one of the four above, you can then add nothing, 256, -256 or other negative numbers to it. This changes the behaviour of the item:

- If nothing is added, different actions result depending on whether the item is "hit" or is "done": when the item is "done", the program comes back from reading the pointer (as we shall see later) but if you only "hit" the item, the item will only change status beteween selected and available a(and back) and that is all.

- If you add 256, the item, even when it is "hit", will cause a return from the read pointer loop, as if it was done. Thus, there is no difference between hitting and doing (!). Also, the item's status is immediately reset to available.

- If you add -256, a hit and a do are, again, the same, but the item is not reset immediately to available.

- You add other negative numbers, but only to text items. If you do that, you will cause a letter in the item (if it is a text!) to be underlined automatically. This is covered in more detail a bit later.

-› * **lstr$,lspr,lblb,lpat** are the arrays containing the content of the items: lstr$ contains strings, lspr contains pointers to sprites, lblb points to pointers for blobs, lpat points to addresses for patterns (we shall see the definition of blobs and patterns later - they are very seldomly used for loose items). Thus, if you have determined (by type%) that the first item is a string, the first element of lstr$ contains this string.

The arrays for these pointers are DIMmed to DIM (n), with lstr$ being dimmed to (n,max_length_of_string) as is usual for string arrays. They parsed - and must be filled - for each corresponding item, as referred to by the ltyp% of the item. Let's suppose we want 3 items, the first one a text item ('HELP'), the second one a sprite items (window move), the third again a text item ('ESC') and the fourth another sprite. We will then DIMension the ltyp% array for four elements: DIM ltyp% (3). The contents of ltyp% will be:

ltyp%(0)=0 (string)
ltyp%(1)=2 (sprite)
ltyp%(2)=0 (string)
ltyp%(3)=2 (sprite)

We will then DIM lstr$(3,10), lspr(3),lblb(3) and lpat(3).

lstr$(0) will contain "HELP", lstr$(2) STAYS EMPTY (0 string), lstr$(2) will be "ESC" and lstr$(3),stays empty again.

lblb and lpat will remain empty (all elements set to 0). Likewise, lspr will be empty except for lspr(1) and lpsr(3) which will each contain a pointer to a sprite (as explained in the last instalment of this series).

The MK_LIL will automatically choose the correct items from the correct arrays, depending on the type of the item. This can be one the worst problems with the QPTR function, i.e. fill in these arays wrongly...

Next time, we deal with automatic underlining of a letter in a text item and information sub-windows.

New Q-Word Game coming soon

Phoebus Dokus informed us about a new project from RWAP, Geoff Wicks and himself. Here is a short description, and have a look at the screen shots! As it is supposed to be ready for XMas, that's the last chance to report about it before its release. We hope to have a review for you as soon as the product is released (hint, hint, for both RWAP and reviewers!)

Q-Word is a word puzzle game that's a fusion between Tetris, Scrabble and your Sunday newspaper's "Find the hidden words" puzzles. Q-Word runs on hi-resolution, hi-colour screens and its the first QL commercial game of its kind to use digital sound (as either a CD sountrack on QPC-QXL or via SSS Q40/Q60/Amiga). There is also planned support for Q-Midi (Via the NET ports on regular QL/QXL and Aurora and in the future via serial on all platforms or Standard Midi UART on uQLx). Q-Word is based on the Q-Typ dictionary, therefore it practically can be used in all languages a Q-Typ dictionary exists. The follow-

Q Branch

Text 87

£ 79.00

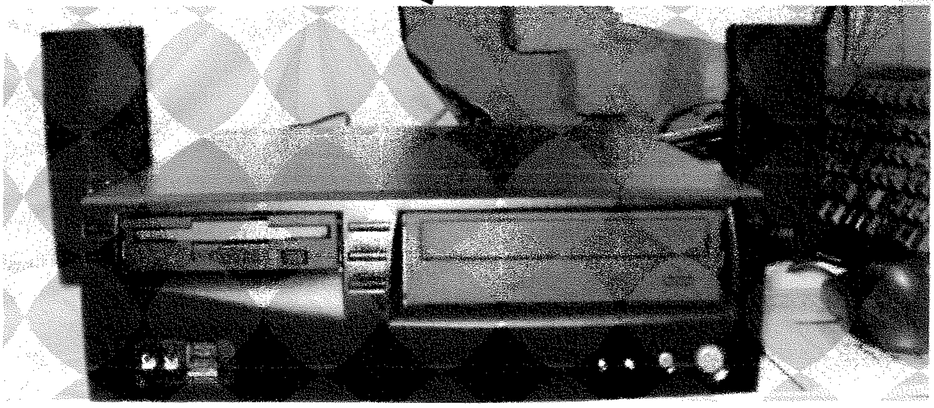
Typeset 94	£ 29.00
Fountext 94	£ 39.00
2488 drivers	£ 29.00
Espson ESC/P2 Drivers	£ 26.00

Programming

QD 98	£ 45.00
QD + QBASIC	£ 59.00
QD, QBASIC, QLIB	£ 100.00
QLiberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 30.00
Easyptr pts 1 & 2	£ 30.00
Easyptr pt 3	£ 14.00
QMake	£ 15.00
QMon / JMon	£ 22.00
Basic Linker	£ 19.00
DISA 3	£ 31.00
QMenu	£ 14.00

CHRISTMAS CHEER

New At Q Branch



The All Black QPC2 Platform,

All For

£ 465.00

Without QPC 2 £ 400.00

Without QPC2/ Windows XP £ 316

The BlaqBox !

A Complete PC with Windows XP and QPC 2 Pre-installed.

You Get :

667 MHz Processor, 256 Mb PC 133 RAM, On-Board VGA (up to 32Mb), On-Board stereo sound (input and output), 2 x USB 1.1 ports, 1x Serial Port, 1 x Bi-Directional Parallel Port, 1 x 10/100 Ethernet port, 56K Modem.

The system comes complete with a black keyboard, mouse and speaker set and includes a free Office Program Suite. I year R.T.B. guarrantee.

'Just Words' - by Geoff Wicks

£ 10.00 ea / 2 programs £ 18.00 / 3 programs £25.00

Includes both P.E. and non-P.E. versions SMSQ/E or P.E. Files needed to run.

QL to PC Convert text files between PC & QL formats

QL Rhymes QL Rhyming Dictionary

QL Thesaurus & QL Style Check

Utilities

FiFi 2	£ 18.00
QSup	£ 28.00
QSpread 2001 v3	£ 48.00
Cueshell II	£ 15.00
Qload / QRef	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 28.00

WE WISH YOU ALL A MERRY CHRISTMAS AND A HAPPY NEW YEAR

Tel +44 (0) 1273-386030 fax +44 (0) 1273-430501
Mobile +44 (0) 7836-745501
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

Q Branch

Feeling out on a limb ?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

The ProWesS word processor

¶ Demo version £ 1.50 + postage
Full Registered version £ 18.00

Version 2.03 available now !

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

QXL II	**£ 50.00**
Recycled superHermes	£ 65.00 *
Recycled Gold Card	£ 45.00 *
Recycled Aurora	See over*
Qubide	See over
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

Qubide upgrades to version 2.01 £ 8.00

SMSQ/E

Gold Card / Atari / QXL Version

Now Only £ 32.00

Various Atari versions : call for details

QPC 2 v3 full colour version!
Now Only £ 65.00

Upgrades from QPC 1 £ 46.00 (return master disk)
Upgrades from QPC 2 v1 £ 30.00
Upgrades from QPC 2 v2 £ 14.00
If Bought/upgraded after 15/10/01 £ 5.00

Special offer !
Get Cueshell for only £15.00 with any copy of SMSQ/E or QPC 2 upgrade !

Q Branch Programs

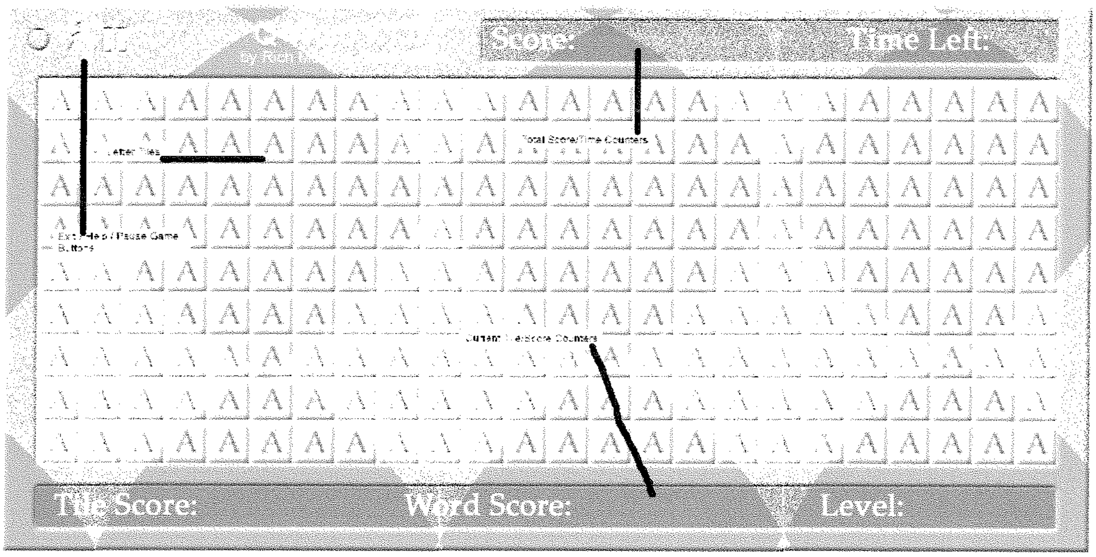
The Knight Safe 3	£ 35.00	Q - Route v1.08C Route finding programme	£ 25.00
upgrades from previous versions	£ 5.00	The Fractal Collection	£ 35.00
Q - Count Pointer driven home accounting	£ 25.00		

Post and Packing is included with Software Prices

VISA

We can accept payment by Visa, Mastercard and Switch. We also accept cheques drawn on a UK Bank in Sterling.
You can also pay directly into our bank in Euros.
Barclays Bank Acc# 79016888 sort code :20-71-02 Name :QBranch)

MasterCard


Score:
Time Left:
Letter Tiles
Total Score/Time Counters
Exit/Help/Pause Game Buttons
Current Tile Score Counters
Tile Score:
Word Score:
Level:

ing supported systems (modes and resolutions - or available colours - are in parentheses) are planned:

Expanded QL
(Mode 4)
Thor
(Mode 12)
Amiga
(Mode 64)
Aurora
(Mode 256)
QXL/QPC
(Mode 32, 640x480 and 800x600)
Q40/Q60
(Mode 33, 512x256 and 1024x512)

The last three entries (Qx0, QXL/QPC and Aurora) are the ones that will appear first for obvious reasons.

Q-Word has been written by the following:
Main Code: Rich Mellor
(RWAP Software)
Word logic/Dictionary support: Geoff Wicks (Just Words)
Graphics and Sound: Phoebus Dokos (Quantum Leap)
Q-Word will be (hopefully) available in December so get ready for Santa Qlaus

For questions and inquiries regarding Q-Word, contact Rich Mellor.
Note: RWAP has moved - see News page!

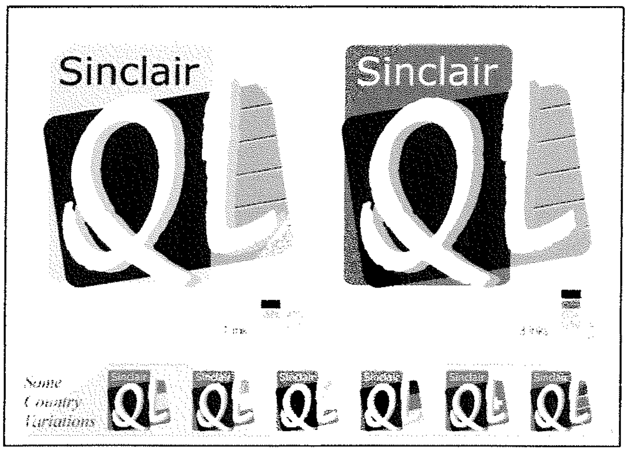
Pity we can't print QL Today in colour - we have received a colour and a grey screenshot, and the colour one looks extremely good!

QL Logo Part II

Dilwyn Jones

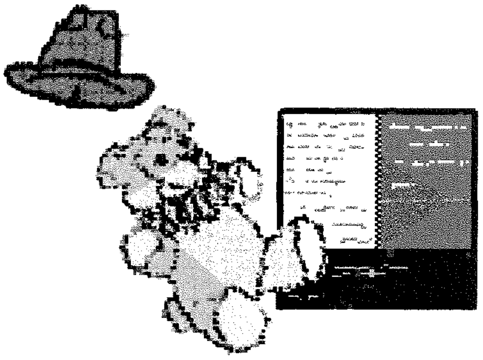
Following the QL Logo article in QL Today Volume 7 Issue 3, I received a couple of further contributions from Javier Guerra and Per Witte.

Javier's offering is a QL Logo with a few international flavours:



Again, colour printing would help a lot here too!

Per Witte's logo offers a little touch of humour, and he makes the remark that it "illustrates the shock of the QL community being overtaken by all those inferior computers."



Feedback on the original article indicates that the favourite so far is the golden 3D logo from Melissa Ward, especially from those who have seen it in colour on my website. Howeverthe number of "votes" received is not really large enough to declare that one a clear winner. Who knows, Per's teddy bear may yet win!
If you want to see all these logos in colour, visit the "QL Logo" web page on my new website at: **http://www.homepages.tesco.net/dilwyn.jones/qllogo/qllogo.html**

BYTS OF WOOD

SAW POINTS OFFCUTS AND SNIPPETS

Bred and Board

In a perfect example of the kind of synchronicity that ties things together I noticed that the QL Users Internet group was buzzing about Dave Park's suggestion that we build a new board for a QL platform based upon some of the smaller chips in PDAs (Personal Digital Assistants). This would be something for which some members of the community would probably pay a bit to own but the user base would be even smaller than that for the Q40/Q60.

Small QL board, as envisioned by Dave, would need to use the kind of chips which are mounted directly onto the board and the cost of setting this process up would be prohibitive for the small number of units involved.

There are, however, other areas in which the small format QL is a possibility. I have long wondered if it would be possible to develop a version of QPC2 for Windows CE, the cut down version of Windows that runs on many of the PDAs. I know that the screen size would be problematic and that much of the software that we use assumes a minimum of 512 x 256 for display but these are software considerations and, although time consuming and brain taxing, somewhat more feasible than a whole new hardware platform.

Another Dream?

One article on 'The Register' (the IT website that I have mentioned before in these pages) struck me as touching on the fantasies of some QL Users with regard to introducing SMSQ/E to a wider audience.
They were talking about the chip manufacturer VIA and their attempts to push the mini ITX format into public prominence. Several of these tiny boards are around and the moment and, by the time you read this, there should be a couple more out. The boards themselves are complete PCs on a plate. Each one has CPU Sound, Video, Serial, Parallel, USB, and Ethernet connectors on board. The new ones which will be arriving soon will also have TV-Out and Firewire ports. This only leaves the user to add RAM and rotating devices to get a complete system. Even the Modem is supplied as an add on board.

To add to the attraction the boards only measure 170mm x 170mm and the lowest version in the range is fanless! Keith Mitchell is trying to put one of these into a MinisQL case at the moment. The only drawback with the two versions currently available is the lack of a floppy port - more a problem for QL Users than PC ones although the floppy drive does have a toehold still in the PC fraternity. The new 'M' series of Via boards have replaced the Floppy Interface.

One manufacturer of small PCs, Shuttle, have shown part of its range at recent Computer Industry shows. The later Via boards have DVD decoding and media player software built into the BIOS and that allows the user to insert a DVD disk and have instant playback via their TV or monitor. A similar process is available for CD playback.

The article went on to say that the only real minus to the whole thing was that, if the user wanted to do anything computer related, he had to wait for Windoze to boot. This, therefore, is the crux of my cunning plan. How about getting SMSQ/E to run native on one of these systems? The whole system as well as most of the software would fit onto a 128Mb compact flash card which could be easily connected to the IDE slot. This would give an almost instant boot and have word processing, spreadsheet and other utility software available.

You have the sources now - what are the chances of that happening?

The BlaqBox

All of the above also led me to consider the slightly larger 'Flex ATX' format PC boards that are around. Since the company that I work for sells the black cases and black keyboard, mice and speaker sets I decided to build one of these units for the London show. (an article about the building of this will be either in this issue or the next depending on the space available). I was surprised to find that a reasonable performance small black PC complete with Windoze XP, a free office suite, and QPC2 could be produced for as little as £465 pounds (without monitor).
A new product for Q Branch I think.

Flying Down To Austria

Once again I have cause to thank Peter Fox for his kindness in Flying down to Austria for the German show. Darren Brannagh was supposed to attend but he was unable to get a cheap flight from Ireland and, due to his forthcoming parenthood he reluctantly decided he had to pull out. Steve Reyal, his usual sidekick at these shows, was able to attend and seemed to have good fun sitting up front in the co-pilot's seat.

The weather at Salzburg, the closest airport to Berchtesgaden, was not too good but it was only a short drive to the hotel where the show was due to be staged. Unfortunately the show had been moved to another venue so we had to go off hunting for a different hotel. This was not too hard even though it was set back off the road and the accommodation was excellent.

The show itself was not that busy although I did have quite a few interesting conversations. One thing that did strike me was the number of people who were using QPC2 on a laptop. There were no actual QLs in evidence, sign of the times I suppose, but no-one, if I recall correctly, even had a desktop machine of any kind.

This may say something about the cheapness of laptops these days and it is certainly easier to travel with a laptop than a tower case, monitor and all the other peripherals.

The food at the hotel was good and we had a convivial evening after the show chatting around the table. I look forward to the next one.

Apathy what Apathy?

Mentioning the laptop situation at the German show leads me on (doesn't everything?) to my task in the last column. I did ask if people would like to write or email me with their preferences for their top three favourite systems. What I wanted was a list of three systems that they own and use with the most favourite/most used system being number one and so on.

Maybe the instructions were too complex for some of you or maybe you all nodded off while reading the last column (I would not blame you) but I received a stunning total of one vote (Thank you John Gregory). I have left this open for another two months to see if I can get any more responses from the readership.

You can email me on **vote@qbranch.demon.co.uk** or write in to the Q Branch address (postcard please - how about one of your town?).

Come on now - I really am interested in how you all feel about your systems.

Byt of Typ

One problem I have had with QPC's use of the PC DOS directories is in remembering which ones I have called by which letters. For instance, on my machine, I have the following:

DOS1 = C:\
DOS2 = D:\QL UPLOADS
DOS3 = D:\QL DOWNLOADS
etc.

Now, if you use QPAC2 to copy files into or out of these directories all you see for a header is DOS1_ or something like that. I have often temporarily forgotten which is which. I did hit on a crude but effective way around this problem. I have just put a dummy subdirectory in the the actual directory with the name of the actual subdirectory in it.

i.e.

DOS2_ is as above but also has a directory called QL UPLOADS in it like this
D:\QL UPLOADS\'QL UPLOADS'

This way when you go to DOS2_ in QPAC2 you get the title 'QL UPLOADS' appearing in the directory select window. As I said crude - but it works. Does anyone have a better idea. Maybe Marcel could use the way that QL disks can have a name in the top right hand corner to reflect a name given in the QPAC2 Configuration. Maybe this is not possible.

Print this Kyocera

Those of you who are frantically trying to get hold of printers that still work with the QL might look into the Kyocera FS 1010 laser printer. It is only monochrome I am afraid but it does work comfortably with the such old time classics as the PSION suite and other programs because it has the EPSON LQ 850, IBM Proprinter and Diablo 630 emulations as well as accepting postscript. This is an excellent low cost laser printer (around £250 currently).

I stumbled across this while installing one in a dentist's surgery. I had read that it had the Epson emulation and I had my laptop with me so I plugged it in and off it went printing from Exchange. I did not even have to install the drivers for Windoze - although it would not have printed from Windoze with them. I just set QPC2 to print to the PAR port.

There did seem to be an issue with the '£' sign which produced a '#' symbol and two hearts but I am sure that, given time, this could be sorted out with a few translates.

Kyocera offer a good two year warranty and gaurantee the drum for life so they are one of the cheapest printers to run. Something for your Christmas list maybe.

Christmas Entertainment

As is now customary at this time of year I have compiled a list of Christmas entertainments suitable for Computer users.

First off we have a run of Beatles Films:

The Yellow Subroutine (featuring the 'Blue Screenies' for Windoze users)
A Hard Drive's Knight (The backup version)
Let IT = BE

Then, if you have interactive TV, press F1 and you get
'Help!'

Other Films include:

Back Windows
feat. Windowa Ryder
Vanishing Point(er)

Three Clint Eastwood films:

The Good the Bad and the Changed Medium
A Fist Full of Microdrives
For a Few$ More

and the classic war film

The LONG(word) and the SHORT(word) and the CALL

For dot matrix printer owners we have:
A Ribbon Runs Through It

On the Blockbusters scene:
Star Wars - The Sinclair Strikes Back

Jochen Merz stars in
Lord of the Things

and the sequel
The Two Tower Cases

For those of you who like Musicals:
Call Me Modem

Cats! featuring Tony Tabby

RAMalot

For music lovers here are a couple of Christmas albums by

The Pointer Sisters
Eek-a-Mouse (don't let Atomic Kitten near them!)

This only leaves me to wish you a Happy Christmas and a Preposterous New Year (oh dear the spiel chucker has glitched again!). I hope to see you all in 2003.

Welcome

2003

Another happy QLing year!

Lots of things are going to happen in 2003 - soft- and hardware-wise - some of them were already announced (and shown!) in this issue of QL Today.

Lots of plans are made "in the background" to arrange future shows in such a way, that it makes sure that most QL dealers and QL users can attend the shows.

Although we have no fixed dates yet, we are happy to report that shows in Eindhoven (NL), Hove (UK), Reggio Emilia (I), Berchtesgaden (D) and somewhere in the USA are on our schedule! But - as always - your attendance is important to help keeping the QL shows alive! If you find a chance, come along!